

AD-A094 807

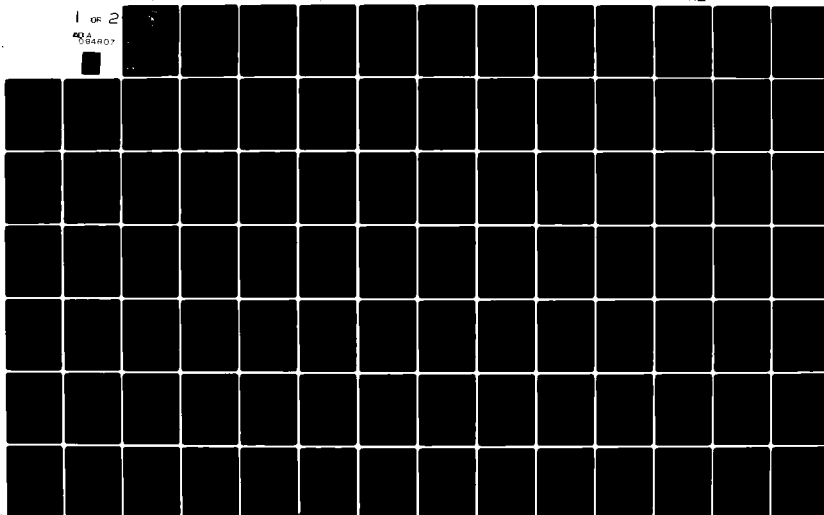
AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOO--ETC F/G 15/7
A GENERALIZED COMPUTER MODEL FOR THE TARGETING OF CONVENTIONAL --ETC(U)
DEC 80 J C PEMBERTON
AFIT/60R/05/80D-6

UNCLASSIFIED

NL

1 OF 2

004
084007

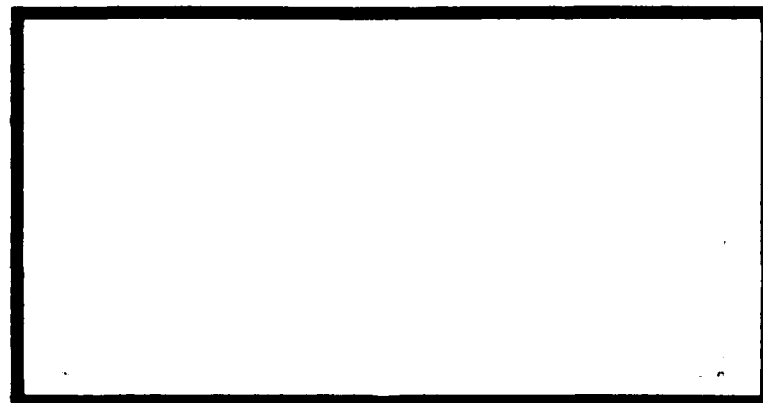


AD A094807



②

DTIC
ELECTE
FEB 1 0 1981
S D



DOC FILE COPY

UNITED STATES AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY
Wright-Patterson Air Force Base, Ohio

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

81 2 09 174

①

FEB 10 1981

A GENERALIZED COMPUTER MODEL FOR
THE TARGETING OF CONVENTIONAL
WEAPONS TO DESTROY A RUNWAY

Thesis

/GOR/OS/80D-6

John C. Pemberton
Captain USAF

Approved for public release; distribution unlimited.

AFIT/GOR/OS/80D-6

A GENERALIZED COMPUTER MODEL FOR THE
TARGETING OF CONVENTIONAL WEAPONS
TO DESTROY A RUNWAY

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science

by

John C. Pemberton, B.S.
Captain USAF

Graduate Operations Research

December 1980

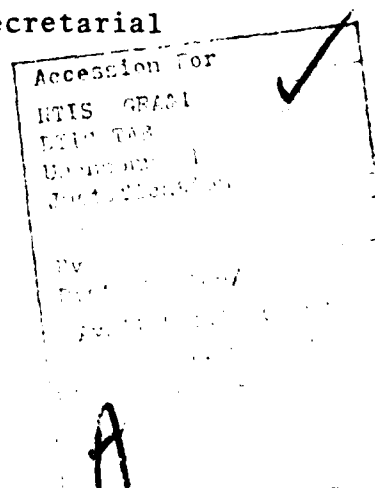
Approved for public release; distribution unlimited.

Preface

This research study was undertaken at the suggestion of Lt Col Edward J. Dunne, Jr., Professor of Operations Research, Operational Sciences Department, School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, Oh. His interest and perception in the problem originated from another study in which a very simplified approach for attacking runways with conventional weapons was attempted as a small portion of the later study.

I have found the area of targeting weapons particularly relevant and applicable to the USAF. By approaching this problem as a group of like subproblems, we hoped to advance this field of study and possibly reduce some of the costs of attacking a runway.

I am grateful to my thesis advisor, Lt Col Dunne, for the assistance, encouragement, and support he provided during this effort. I am also indebted to Prof. D. Wallace Breuer and Major Daniel B. Fox for their understanding and technical advice. Finally, I wish to express my sincere appreciation to Mrs. Suzanne Weber for her secretarial expertise.



Contents

	<u>Page</u>
Preface.	ii
List of Figures.	v
List of Tables	vi
List of Frequently Used Symbols.	vii
Abstract	viii
I. Introduction	1
Background	1
Problem.	2
Assumptions.	2
Scope.	3
Approach and Presentation.	4
II. Reduction of the Two Dimensional Problem to a One Dimensional Problem.	5
Introduction	5
Transformation of the Independent Variables.	5
Effective Explosive Weight	6
Circular Error Probable.	6
Actual Runway Length and Width	7
Runway Destruction Criteria.	8
Subdividing the Overall Runway Problem	8
Converting the Subproblem to One Dimension	12
Summary.	13
III. Methods of Approximating the Probability of Achieving a Cut.	14
Introduction	14
Approximation Methods for a Given Aim Point Strategy	14
Order Statistics Method.	15
Event-Composition Method	15
Monte Carlo Simulation	16
Description of the Event-Composition Method.	16
The Numbers of the Events Which Comprise E	17
Probability of Event A_{jki}	19

	<u>Page</u>
Probability of Event L_i	19
Probability of Event E_i	20
Solving for $\bar{P}(A_{jki})$	22
Description of the Monte Carlo Simulation	
Method.	23
Summary	23
IV. A Search Routine for Determining an Aim Point	
Strategy.	25
Introduction.	25
Determining the Initial Bounds on the Required	
Number of Weapons	26
Determining the "Best" Aim Point Locations for	
Each Aim Point Set Within an Aim Point Group.	27
Determining the Best Aim Point Set Within an	
Aim Point Group	32
Methods for Decreasing the Search Algorithm	
Calculation Time.	36
Summary	37
V. Verification and Validation of the Model.	38
Introduction.	38
Verification.	39
Verifying the Two Approximating Methods for	
PC.	39
Verification of the Time Saving Methods	44
Factorial Design.	45
"Validation" of the Model	48
Summary	51
VI. Conclusions and Recommendations	52
Conclusions	52
Recommendations	52
Bibliography	54
APPENDIX A: Derivation of the Effective Explosive	
Weight Conversion Formula	56
APPENDIX B: Derivation of the Circular Error Probable	
Conversion Formula.	59
APPENDIX C: FORTRAN Coding for the Model.	62
APPENDIX D: Derivation of the Mean and Variance of $\hat{\bar{P}}$,	
the Estimated Probability of Destroying	
the Runway with the "Best" Aim Point	
Strategy.	100
Vita	104

List of Figures

<u>Figure</u>	<u>Page</u>
1 Minimum Takeoff Distance (D)	9
2 Interval Between Centerlines of Two Adjacent Cuts (B).	9
3 Interval from the Leading Edge of the Actual Runway to the Trailing Edge of the First Cut.	12
4 A Graphic Representation of Events A_{jki} , L_i , and \bar{E}	17
5a Best Locations for a Set Within a Group. . . .	28
5b Best Locations for a Set Within a Group. . . .	29
6 Best Aim Point Set Within an Aim Point Group .	33
7 The Best Aim Point Strategy.	35
8 Comparison of $P(\bar{E})$ to $P(E)$	41

List of Tables

<u>Table</u>		<u>Page</u>
I	Comparison of \overline{PC} Approximation Methods	43
II	Results from the 2^3 Factorial Design	47
III	ANOVA of the Main Effects of W, CEP, and V on the Number of Weapons Required to Make a Cut.	48

List of Frequently Used Symbols

E	the event that at least one minimum launch window's width remains undamaged at the completion of an attack
EX	the effective runway length
EY	the effective runway width
M	the number of simulated attacks
P	the runway destruction criterion
PC	the destruction criterion for a subproblem
P(E)	the probability of event E occurring
PS	the probability of achieving a cut with a specific aim point strategy
R	the damage radius caused by the weapon employed
σ	the standard deviation of the distribution of a weapon's impact points about its aim point
U	the minimum launch window's length
V	the minimum launch window's width
W	the effective explosive weight of the weapon employed
X	the runway length
Y	the runway width

Abstract

A generalized computer model for determining where to target conventional weapons against a non-reinforced concrete runway is developed. This development consists of taking the overall problem, runway destruction, and dividing it into a number of identical, stochastically independent subproblems. Then one of these subproblems is used to determine an aim point strategy which reduces the number of weapons required to achieve the level of runway destruction desired.

In order to solve the subproblem, a search algorithm is developed. Along with this search algorithm, two methods for approximating the level of destruction, given an aim point strategy, are investigated. Finally, the entire model is verified and "validated" by a number of statistical tests. Also a comparison of this model to the Airfield Assessment Program, an Air Force Armament Laboratory model, is provided. Using the model's results for the data investigated, an equation relating circular error probable and minimum launch window's width to the number of weapons employed is derived.

A GENERALIZED COMPUTER MODEL FOR THE
TARGETING OF CONVENTIONAL WEAPONS
TO DESTROY A RUNWAY

I. Introduction

Background

The airfield complex, normally attacked in hopes of reducing an enemy's air capability, has been a popular target. Of primary importance on the complex have been the aircraft themselves because of their delicate structure, fuel content, and high-explosive weapons. Two factors have made aircraft a less opportune target and centered the attack on airfield pavement surfaces. First, hardened shelters have made the aircraft harder to destroy. The attacker must breach or substantially damage the shelter to reach the aircraft. This requires special weapons whose accuracy demands aerial maneuvers which make the attacking planes vulnerable to ground defenses. The second factor has been the increase in aircraft performance and the consequent intensity with which the first stages of a war will be fought. Attacking planes can strike, return to base, rearm, and be over the target again in an extremely short time, often shorter than that required to repair a runway and launch interceptors. Opposition fighters can thus be effectively eliminated from the first stages of battle without being physically damaged by denying them a launch and recovery surface. The concept that the runways. . . are poor targets is no longer valid in the light of the pace of modern warfare and the capabilities of aircraft [Ref 7: 260].

Currently, there is extensive research being conducted in developing conventional weapons for runway destruction, but there seems to be no readily available targeting scheme on how best to employ these weapons. There are a number of computer models used by the Air Force to simulate runway destruction; for example, the Airfield Assessment Program,

AAP (Ref 3), but these models require the user to supply the aim points he desired. The models then determine, through Monte Carlo simulation, whether the runway has been sufficiently destroyed so a specified minimum launch window no longer exists (minimum launch window being the minimum length and width of runway required for takeoff or landing). The main drawback to these investigations is the locations of the aim points and the number of weapons targeted per aim point (referred to for the rest of this thesis as the "aim point strategy") are mostly derived from the experience of the user and his best guess as to what this strategy should be. If one could determine an aim point strategy which reduces the total number of weapons employed and still maintain the desired level of runway destruction, then one could lower the overall costs without lowering the effectiveness (assuming that all other costs were equal).

Problem

The problem is to develop a computer model which will determine an aim point strategy that achieves the desired level of runway destruction with as few total weapons as possible.

Assumptions

1. The circular error probable, CEP, of the weapons employed can be logically described by a circular normal probability distribution.
2. The survivability of all weapons is one. As long

as the weapon's survivability is the same for all aim points, this assumption will only reduce the total number of weapons employed and will not affect the number and the locations of the aim points.

3. The reliability of all weapons is one.

4. Each weapon is independently targeted so that its impact point does not depend on any other weapon's impact point or aim point.

5. Only one type of weapon can be used to destroy the runway. Thus, the CEP and the effective explosive weight will remain constant throughout an investigation (investigation being the determination of a good aim point strategy for a specific set of inputs).

6. The time frame between the launching of the first weapon and the launching of the last weapon is too short to update aim points through damage assessment.

7. Any minimum launch window must be totally composed of actual runway.

8. If a minimum launch window exists, it is assumed to be accessible to an aircraft, even though this may not be true (Ref 3).

9. The runway destruction criteria is a given confidence level that no minimum launch window will remain undamaged after the attack.

Scope

Aside from the limitations inherent in the assumptions, the model is designed for investigations using conventional

weapons in destroying non-reinforced concrete runways.¹ Yet the model is general enough to investigate any size non-reinforced concrete runway and any size minimum launch window.

The model was used to investigate destroying a runway 8000 feet long and 150 feet wide (the dimensions considered typical of a North Atlantic Treaty Organization runway (Ref 3)) and a minimum launch window 2000 feet in length (a length which is slightly shorter than a typical fighter's takeoff roll (Ref 8)). This minimum launch window's length was coupled with a width of either 50 feet or 100 feet. With these sets of dimensions, effective explosive weights of 250, 500, and 1000 pounds and CEPs of 20, 50, and 100 feet were investigated. The runway destruction criterion for all investigations was maintained at a level of .80.

Approach and Presentation

The approach used in this thesis is first to reduce the two dimensional problem to a set of independent, identical one dimensional subproblems (Chapter II). Second, for a given aim point strategy, two methods for approximating the destruction level are developed (Chapter III). Third, a search algorithm which determines a good aim point strategy is developed (Chapter IV). Next, the two approximation methods and the algorithm are combined for verification and validation (Chapter V). Finally, a discussion of the conclusions and recommendations for further investigation is presented (Chapter VI).

¹Atypical of current runways (see Recommendation 5)

II. Reduction of the Two Dimensional Problem to a One Dimensional Problem

Introduction

As was pointed out in Chapter I, the approach used in this thesis is to reduce a two dimensional problem (the two dimensions being length and width) to a one dimensional problem (just width). In order to accomplish this reduction, the independent variables (the "givens" for the problem) must first be transformed into a new set of variables. Next, the overall runway problem will be subdivided into a number of like, smaller, stochastically independent subproblems (the runway "cuts"). Since the subproblems are alike and stochastically independent, by minimizing the number of weapons required to accomplish any one cut, the minimum number of weapons for any cut is then determined. Finally, the two dimensional subproblem (any one cut) is transformed into a one dimensional subproblem.

Transformation of the Independent Variables

The five independent variables the user is required to provide are:

1. The weapon's effective explosive weight, W , in pounds TNT
2. The weapon's circular error probable, CEP, in feet

3. The actual runway length, X, and width, Y, in feet
4. The length, U, and width, V, of the minimum launch window
5. The runway destruction criterion, P (a real number between zero and one which represents the desired confidence level that no minimum launch window remains undamaged after the attack)

Four of these independent variables (W, CEP, X and Y, and P) are converted into a new set of independent variables (damage radius, standard deviation, effective runway length and effective runway width, and the cut destruction criterion).

Effective Explosive Weight. The effective explosive weight is converted into the damage radius, R, this explosive would cause in non-reinforced concrete runways. This conversion is accomplished by determining the optimal depth of burst (DOB) for a specific weapon, the damage radius caused by this specific weapon at its optimal DOB, and a scaling factor. Coupling these three items, one can derive the following equation (see Appendix A for the derivation):

$$R = (3.54)(W)^{1/3} \quad (2-1)$$

Circular Error Probable. The second independent variable transformed is the circular error probable, CEP, for the weapon. CEP is the radius of a circle which encloses 50 percent of the weapons' impact points given that the weapons are targeted for the center of this circle and there are no biasing errors. For example, if ten weapons with identical CEPs of

50 feet are targeted at an aim point, on the average five of these weapons will impact within 50 feet of that aim point. The approach used by this thesis required that CEP be converted into the standard deviation (σ) of the weapon. The conversion factor is

$$\sigma = \frac{CEP}{(2 \ln 2)^{1/2}} \quad (2-2)$$

(see Appendix B for the derivation).

Actual Runway Length and Width. The third transformation is from actual runway length and width to effective runway length and width. The effective runway dimensions are defined to be those dimensions within which a weapon can impact and cause damage to the actual runway. The effective runway's dimensions will be greater than those of the actual runway because a weapon can impact off the actual runway up to a distance of R and still cause damage to the runway. The effective runway length, EX, is

$$EX = X + 2R \quad (2-3)$$

and the effective runway width, EY, is

$$EY = Y + 2R \quad (2-4)$$

Using these dimensions, EX by EY, there are four areas in which a weapon could impact and still not cause damage to the actual runway. These four areas are near the corners of the rectangle formed by EX and EY. Even though these areas do exist, when compared to the entire area under investigation, they are insignificant and will be ignored.

Runway Destruction Criterion. The final transformation is to convert the runway destruction criterion, P, to the destruction criterion, PC, for one of the subproblems. As a first cut, it is assumed the subproblems are identical and stochastically independent, thus

$$PC = (P)^{1/N} \quad (2-5)$$

where N is the number of subproblems. (Note: when one uses whole numbers of weapons to achieve PC and requires each subproblem to be identical, there may be some "overkill" when the subproblems are combined into the final solution.)

Subdividing the Overall
Runway Problem

One now has the following set of independent variables:

1. The damage radius, R, of the weapon
2. The standard deviation, σ , of the weapon impact pattern
3. The effective runway length, EX, and the effective runway width, EY
4. The length, U, and width, V, of the minimum launch window
5. The destruction criterion for each of the subproblems, PC

The approach used in this thesis is to cut the runway across the effective width in enough locations so lengthwise the runway destruction criterion is achieved. The size of each of these weapon impact areas, cut, will be 6σ by EY (see Fig 2).

6σ is chosen as the cut's width to help simplify the subproblem. If a weapon is targeted for the centerline of a cut, one is 99 percent confident that the weapon will impact

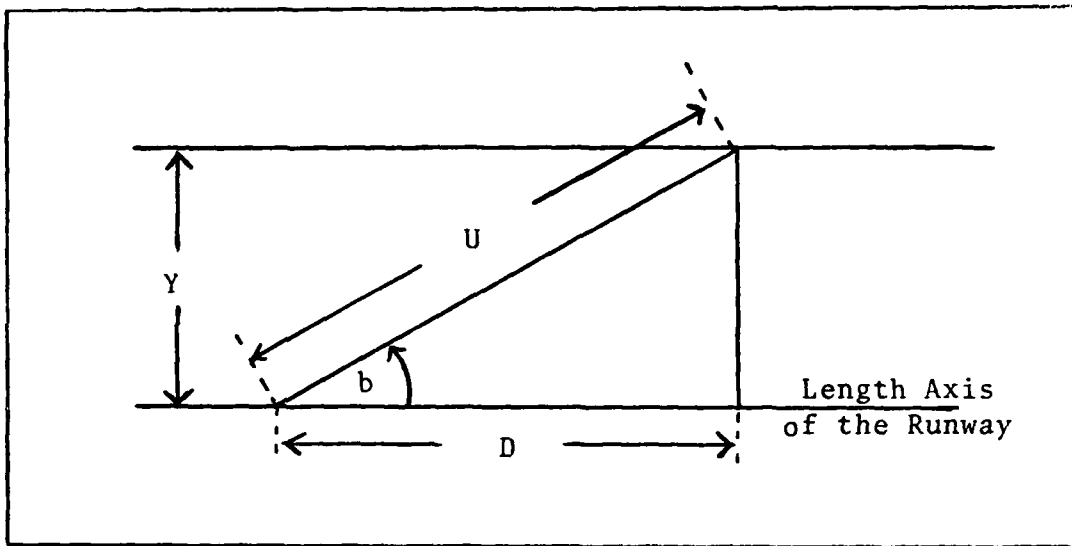


Fig 1. Minimum Takeoff Distance (D)

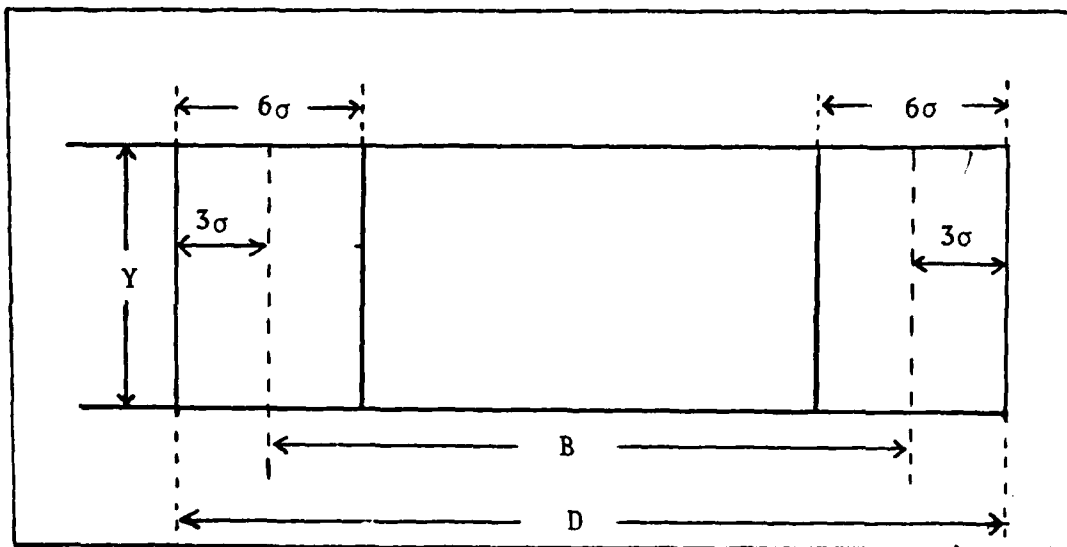


Fig 2. Interval Between Centerlines of Two Adjacent Cuts (B)

within 3σ of the centerline. Thus, the probability that this weapon impacts further than 3σ from the centerline is extremely small, and will be assumed to be zero. In addition to the cut dimensions, one must determine the minimum takeoff distance along the runway's length. If an aircraft takes off across the runway rather than along the axis of the runway's length, it would require less of the actual runway's length. By knowing U and Y , the minimum takeoff distance (along the length axis of the runway), D , is (see Fig 1):

$$D = (U^2 - Y^2)^{1/2} \quad (2-6)$$

Even though this type of takeoff violates Assumption 7 (Chapter I), D is less than any minimum takeoff distance required under Assumption 7. Thus, if the runway destruction criteria is met using D , it is also met under Assumption 7.

D can now be used to determine the interval between any two adjacent cuts on the runway. Coupling the dimensions of the cut with D , the interval between the centerlines of any two adjacent cuts, B , is (see Fig 2):

$$B = D - 6\sigma \quad (2-7)$$

(An implicit assumption in this approach is that no two cuts will overlap. If they do overlap, the cuts are no longer stochastically independent, and PC must be computed using conditional probabilities.) B is appropriate for any two adjacent cuts, but it is inappropriate for the leading edge of the actual runway and the trailing edge of the first cut.

Leading will refer to the left extreme of the object associated with it; for instance, the actual runway. Trailing will indicate the right extreme of the object it is associated with; in this case, the first cut (as indicated in Fig 3). The interval between the leading edge of the actual runway and the centerline of the first cut (the first interval) is equal to $D - 3\sigma$. Thus, the total number of cuts, N , to be made in the runway is

$$N = \left[\frac{X - (D - 3\sigma)}{B} \right] + 1 \quad (2-8)$$

where the symbol $[a]$ indicates the largest integer value less than or equal to a . N must be an integer value because any remainder can be viewed as if it is the distance from the leading edge of the last cut to the trailing edge of the actual runway. This distance must be less than D , and thus it is less than the length of the minimum launch window. The $+1$ on the right-hand side of Eq (2-8) accounts for the first cut which is being subtracted from X in order to determine the number of remaining cuts (D is the interval between any two adjacent cuts, and this interval is less than the first interval).

If the N cuts are positioned at their proper intervals along the runway, any minimum launch window must pass through at least one of these cuts (Assumption 7 forces this to be true). Also, the angular difference between a minimum launch window along the length axis of the runway and a minimum launch window along any axis which satisfied Assumption 7 (an angle

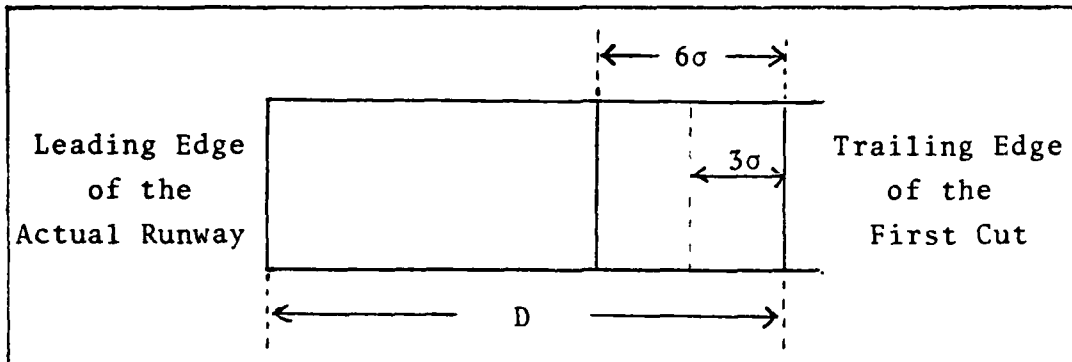


Fig 3. Interval from the Leading Edge
of the Actual Runway to the
Trailing Edge of the First Cut

less than angle b in Fig 1) is very small. One can thus assume that if PC is met for an axis along the length of the runway, PC is effectively met for any axis. Furthermore, if all cuts are achieved at a level of PC, the actual runway destruction criteria is met; this discussion should justify Eq (2-5).

Converting the Subproblem to One Dimension

By combining two previously discussed ideas, the two dimensional subproblem can be transformed into a one dimensional subproblem. These two ideas are:

1. If PC is met for any axis along the length of the runway, then PC is effectively met for any axis.
2. Any minimum launch window must pass entirely through one cut.

Thus, all impact points within a cut may be viewed as if they

are on the cut's centerline. Also, due to the assumption used in choosing the cut's width, the circular normal probability distribution can be viewed as a normal probability distribution in one dimension. Finally, since all cuts are identical in size and PC, one only needs to determine the optimal aim point strategy for a single cut to have the optimal aim point strategy for all cuts (the only real difference is in the locations of the cuts' centerline). Furthermore, due to the exponential form of the cumulative probability distribution for any PC, fewer weapons are required to satisfy P if all cuts have the same probability of destruction. Thus, the minimum total number of weapons required to achieve P is N times the number of weapons needed to make any one cut at a confidence level of PC.

Summary

The overall two dimensional problem has been converted to a one dimensional subproblem. This was accomplished by analytically transforming four of the five independent variables into new independent variables. This new set of independent variables was then used to logically divide the overall problem into a number of like, stochastically independent subproblems in two dimensions. Finally, due to the stated assumptions and the formulation of the subproblems, each subproblem can be viewed in one dimension and by solving only one of the subproblems, the entire problem is solved.

III. Methods for Approximating the Probability of Achieving a Cut

Introduction

Using the set of model parameters and the simplification of the overall problem discussed in Chapter II, the one dimensional subproblem can be expressed as follows:

Determine the aim point strategy which
Minimizes the number of weapons required
Subject to no section, wider than the minimum (3-1)
 launch window's width, remains un-
 damaged with a probability of at
 least PC

In order to develop an algorithm which approximates the solution to this subproblem, a building block approach was used. The first phase was to determine an adequate approximation for the probability of making a cut, PC, given an aim point strategy. This required choosing the approximation methods and then adapting the subproblem to the approximations used.

Approximation Methods for a Given Aim Point Strategy

Three methods were investigated. The first method incorporates order statistics. The second uses the event-composition method. And the third method involves a Monte

Carlo simulation.

Order Statistics Method. Under this method, the constraint can be formulated as follows:

1. Let H be the probability distribution of the weapons' impact locations on the effective runway for a given aim point strategy.
2. Let $h_{(i)}$ be the realization of the i^{th} order statistic, where $h_{(0)}$ is one edge of EY ; $h_{(N)}$ is the other edge of EY ; and $h_{(i)}$, $i = 1, 2, \dots, N-1$, are the impact points themselves.
3. Let G be the probability distribution of the maximum difference between $h_{(i)}$ and $h_{(i+1)}$.

Then

$$P(g \leq V + 2R) \geq PC \quad (3-2)$$

In order to use this method, one first needs to approximate the probability distribution of H . But the distribution of H , in most cases, is the joint distribution of more than one normal probability distribution. Coupling this fact with the requirement that $h_{(i)}$ could be from any of the normal distributions about their aim points for the given strategy, the order statistic method was considered not promising as a model for estimating PC .

Event-Composition Method. Under the event-composition method, the event of interest, the achieving of a cut, is expressed "as a composition (unions and/or intersections) of two or more other events [Ref 9: 42]." This method requires

transforming a continuous probability distribution, I (the probability distribution of achieving a cut with a given aim point strategy), into a set of discrete probability subsets.

Monte Carlo Simulation. With the Monte Carlo simulation, the impact points of the weapons are generated using a normal random generator with a mean of zero and a standard deviation of one. By coupling these random numbers with the aim point strategy and the standard deviation of the weapons' impact points, one can produce a set of random variates which will duplicate an expected attack. The set of variates can then be tested to see if this realization of the aim point strategy cut the runway.

Description of the Event-Composition Method

Rather than approximating PC for a given aim point strategy, the complement of PC, \overline{PC} , was chosen. In order to use the event-composition method, one must first define the events which comprise the overall event of interest.

1. E is the event that at least one L_i occurs in an attack (by definition $P(E) = \overline{PC}$).
2. L_i is the event that no weapon damages the i^{th} minimum launch window's width (this event occurs when no weapon impacts in the i^{th} grouping).
3. A_{jki} is the event that no weapon targeted for the j^{th} aim point impacts in the k^{th} section (length Z) of the i^{th} grouping. (For further understanding of these events, refer to Fig 4)

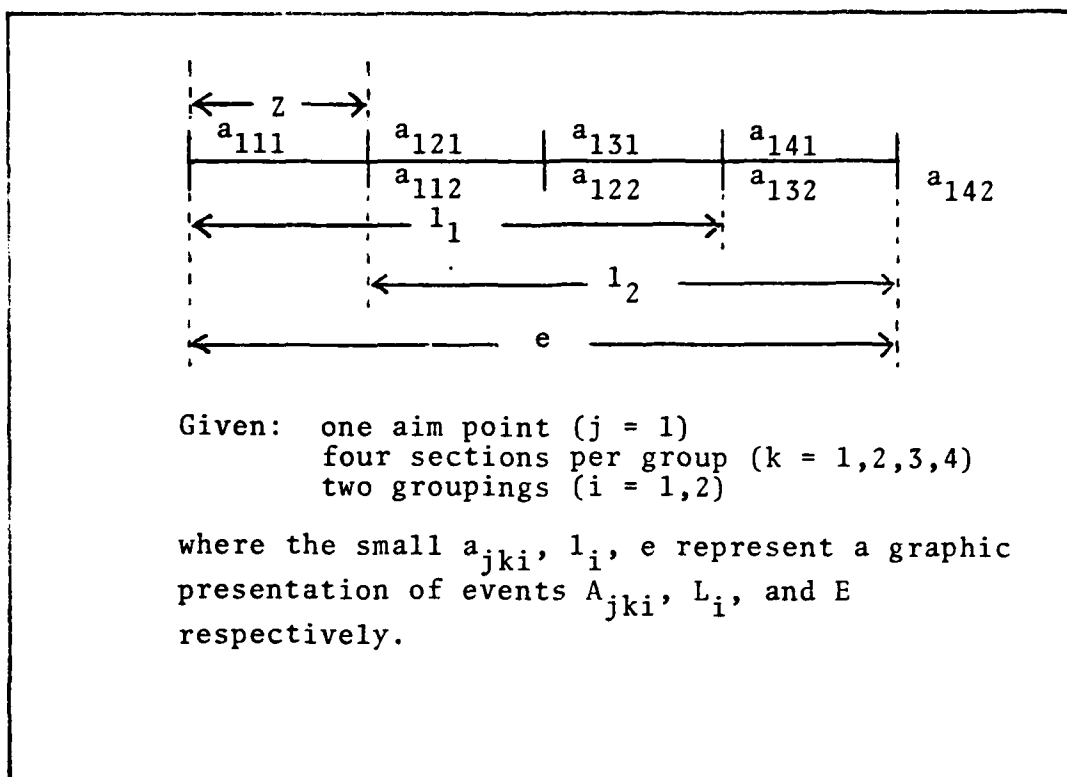


Fig 4. A Graphic Representation of Events A_{jki} , L_i , and E

The Numbers of the Events Which Comprise E . Prior to determining the probability of a specific A_{jki} , $P(A_{jki})$, one must first know the length Z , part of the definition of A_{jki} . In determining Z for a given aim point strategy, there is an inverse relationship between the length of Z and the accuracy of this method in approximating the actual continuous situation, i.e. as Z becomes longer, this method becomes less accurate. Yet as Z decreases, user's inputs remaining constant, the computation time increases.

It is felt the user's minimum launch window's width, V ,

is a point estimate of the actual required minimum launch window's width. This being true, the results for any continuous case based on the user's inputs would, in itself, be an approximation. As a first decision Z_1 was set to .1V. This Z value, Z_1 , is used for all sections within a grouping except for the last section within each grouping.

In order to cause any estimate of failing to cut the runway to be higher than is true, or "optimistic", the following formulas were chosen to determine the number of sections per width, NA; the number of sections per grouping, NL; and the number of groupings per width, NW.

$$NA = \left[\frac{Y+R}{Z} + 1 \right] \quad (3-3)$$

where Y is the actual runway width and R is the damage radius.

$$NL = \left[\frac{Y+R+F}{Z} + 1 \right] \quad (3-4)$$

where F is

$$F = \left(\frac{Y}{R} - \left[\frac{Y}{R} \right] \right) R \quad (3-5)$$

$$NW = NA - NL + 1 \quad (3-6)$$

The +1 on the right-hand sides of Eqs (3-3) and (3-4) allows one to incorporate the fraction, F, into each grouping for a specific L_1 . The last section in each grouping (Z_2 in size), except for the last grouping, is actually shorter than the other sections within the grouping. The last section

the last grouping (Z_3 in size) not only incorporates the fraction, F_i , it also includes the remaining effective runway width not accounted for to this point. Thus,

$$Z = \begin{cases} Z_1 & \text{for } \Psi_{j,i} \text{ and } k = 1, \dots, NL-1 \\ Z_2 & \text{for } \Psi_j; i=1, \dots, NW-1; k=NL \\ Z_3 & \text{for } \Psi_j; i=NW; k=NL \end{cases} \quad (3-7)$$

With the numbers of events which comprise the subproblem, one can use this information in determining the probability of E, $P(E)$. The approach to accomplishing this task will start with A_{jki} , use the A_{jki} 's to build the L_i 's, and then use the L_i 's to build E.

Probability of Event A_{jki} . Each A_{jki} within grouping i are designed to be mutually exclusive (see Fig 4). Yet, they are not independent of each other. If we know that A_{jk1} occurs, then any revised probability of A_{jk2} occurring will decrease. Assuming there are N aim points ($j = 1, \dots, N$) and one weapon per aim point, the probability of A_{jki} , $P(A_{jki})$ is

$$P(A_{jki}) = 1 - \prod_{j=1}^N P(\bar{A}_{jki}) \Psi_{k,i} \quad (3-8)$$

where \bar{A}_{jki} is the complement of A_{jki} .

Probability of Event L_i . By definition L_i only occurs when all A_{jki} within grouping i occur. This can be expressed as follows:

$$L_i = A_{1i} \wedge A_{2i} \wedge \dots \wedge A_{NL,i} \quad (3-9)$$

or

$$L_i = \bar{A}_{1i} \cup \bar{A}_{2i} \cup \dots \cup \bar{A}_{NL,i} \quad (3-10)$$

Since all A_{jk} within a grouping i are mutually exclusive, the probability of L_i , $P(L_i)$ is (Ref 9: 44)

$$P(L_i) = 1 - \sum_{k=1}^{NL} P(\bar{A}_{1,k,i}) \quad (3-11)$$

(Note: this refers to one weapon at aim point 1)

By expanding the number of aim points to N and realizing that events from different aim points are independent,

$$P(L_i) = \prod_{j=1}^N (1 - \sum_{k=1}^{NL} P(\bar{A}_{jki})) \quad (3-12)$$

(Note: still only one weapon per aim point)

Probability of Event E. As previously defined, E is the union of all L_i 's, thus

$$E = L_1 \cup L_2 \cup \dots \cup L_{NW} \quad (3-13)$$

The probability of event E , $P(E)$ is (Ref 9: 46)

$$\begin{aligned} P(E) = & \sum_{i=1}^{NW} P(L_i) - \sum_{a=1}^{NW-1} \sum_{b=2}^{NW} P(L_a \cap L_b) \\ & + \sum_{a=1}^{NW-2} \sum_{b=2}^{NW-1} \sum_{c=3}^{NW} P(L_a \cap L_b \cap L_c) - \dots + \dots - \dots + \\ & - (-1)^{NW} P(L_1 \cap L_2 \cap \dots \cap L_{NW}) \end{aligned} \quad (3-14)$$

Incorporating $P(L_i)$, Eq (3-12):

$$\begin{aligned}
P(E) = & \sum_{i=1}^{NW} \left\{ \prod_{j=1}^N \left(1 - \sum_{k=1}^{NL} P(\bar{A}_{jki}) \right) \right\} - \\
& \sum_{a=1}^{NW-1} \sum_{b=2}^{NW} \sum_{j=1}^N \left\{ \prod_{k=1}^{NL} \left(1 - \sum_{k=1}^{NL} P(\bar{A}_{jkb}) - \sum_{k=1}^{\text{MIN}(NL, b-a)} P(\bar{A}_{jka}) \right) \right\} \\
& + \sum_{a=1}^{NW-2} \sum_{b=2}^{NW-1} \sum_{c=3}^{NW} \sum_{j=1}^N \left\{ \prod_{k=1}^{NL} \left(1 - \sum_{k=1}^{NL} P(\bar{A}_{jkc}) - \sum_{k=1}^{\text{MIN}(NL, c-b)} P(\bar{A}_{jkb}) - \sum_{k=1}^{\text{MIN}(NL, b-a)} P(\bar{A}_{jka}) \right) \right\} - \dots + \dots \\
& - (-1)^{NW} \left\{ \prod_{j=1}^N \left(1 - \sum_{k=1}^{NL} P(\bar{A}_{j,k,NW}) - P(\bar{A}_{j,1,NW-1}) - \dots - P(\bar{A}_{j,1,1}) \right) \right\}
\end{aligned} \tag{3-15}$$

where the symbol $\text{MIN}(a,b)$ means the minimum of a and b .

The only feature which $P(E)$ of Eq (3-15) does not incorporate is the number of weapons targeted for a specific aim point (other than the trivial case of one weapon per aim point). By letting T_j be the number of weapons targeted for the j^{th} aim point, $P(L_i)$ becomes

$$P(L_i) = \left(1 - \sum_{k=1}^{NL} P(\bar{A}_{jki}) \right)^{T_j} \tag{3-16}$$

and thus,

$$\begin{aligned}
P(E) = & \sum_{i=1}^{NW} \left\{ \prod_{j=1}^N \left(1 - \sum_{k=1}^{NL} P(\bar{A}_{jki}) \right)^{T_j} \right\} \\
& - \sum_{a=1}^{NW-1} \sum_{b=2}^{NW} \sum_{j=1}^N \left\{ \prod_{k=1}^{NL} \left(1 - \sum_{k=1}^{NL} P(\bar{A}_{jkb}) - \sum_{k=1}^{\text{MIN}(NL, b-a)} P(\bar{A}_{jka}) \right)^{T_j} \right\}
\end{aligned}$$

$$+ \dots - \dots + \dots - (-1)^{NW} \left\{ \prod_{j=1}^N \left(1 - \sum_{k=1}^{NL} P(\bar{A}_{jk,NW}) \right) - \right.$$

$$\left. P(\bar{A}_{j,1,NW-1}) - \dots - P(\bar{A}_{j,1,1}) \right\}^T j \quad (3-17)$$

(See Appendix C, Subroutine UNION for the coding of Eq (3-17).)

Solving for $P(\bar{A}_{jki})$. With $P(E)$ as formulated, the only probabilities which must be determined to solve Eq (3-17) are the $P(\bar{A}_{jki})$'s. Since the distribution of the weapons' impact points around their aim points is a normal probability distribution, one needs to:

1. Translate EY such that the location of aim point j is assigned a value of zero.
2. Convert the translated leading edge, d , and the translated trailing edge, f , of each a_{jki} into standard deviations about aim point j . In a majority of cases, the trailing edge of a_{jki} is the leading edge of $a_{j,k+1,i}$.
3. Solve the following integral:

$$P(d \leq x \leq f) = \frac{1}{\sqrt{2\pi}} \int_{d/\sigma}^{f/\sigma} \exp - \frac{1}{2}(x^2) dx \quad (3-18)$$

where $P(d \leq x \leq f)$ is the probability a weapon targeted for aim point j will impact such that \bar{A}_{jki} will occur. (See Appendix C, Subroutine PROCAL for the coding of this transformation)

Description of the Monte Carlo Simulation Method

As with the event-composition method, the Monte Carlo simulation method approximates \overline{PC} for a given aim point strategy. As was explained in the paragraph Monte Carlo Simulation, the set of random variates (an "attack") either cuts the runway or it does not cut the runway. If one simulates M , identical and independent, "attacks", then this method becomes a binomial experiment (Ref 9: 72). And

as a rough rule, you can assume that the distribution of \hat{p} (\hat{p} , the estimate of \overline{PC}) will be mound-shaped and approaching normality for (large) sample sizes such that $p \pm 2\sqrt{\frac{p(1-p)}{M}}$ lies in the interval (0,1) [Ref 9: 270].

By applying the asymptotic normality property of a binomial probability distribution, one can determine the number of "attacks", M , needed to have a confidence interval about \hat{p} of $\pm .01$ at a confidence level of .99*.

$$M = \frac{(2.576)^2 p(1-p)}{(.01)^2} \quad (3-19)$$

Yet for the subproblem, $p = \overline{PC}$ thus

$$M = \frac{(2.576)^2 (\overline{PC})(PC)}{(.01)^2} \quad (3-20)$$

*These values were arbitrarily chosen and can be easily changed (see Appendix C, Subroutine ITER).

Summary

For a given aim point strategy, two methods for

approximating \overline{PC} were discussed. First by using an event-composition method, $P(E)$, the probability of not accomplishing a cut, was derived from the unions and/or intersections of its subsets. And second, by the use of a Monte Carlo simulation, one can approximate \overline{PC} by \hat{p} , an estimate of the probability of not accomplishing a cut.

IV. A Search Routine for Determining an Aim Point Strategy

Introduction

Both approximation methods from Chapter III, the event-composition method and the Monte Carlo simulation method, require a given aim point strategy prior to estimating the probability of not making a cut, \overline{PC} . The second phase of this research is to develop methods to determine "good" aim point strategies. This chapter reports on the development of a search algorithm which derives an aim point strategy for achieving the desired \overline{PC} . The algorithm decisions always choose the strategy with the fewest number of weapons, but this final strategy may not be the optimal aim point strategy, i.e. it may not minimize the total number of weapons required. The search algorithm is ultimately based on two assumptions. Given all other conditions are equal:

1. The fewer the number of aim points, the better.
2. Aim point locations which are symmetric about the middle of a cut are preferred to locations which are not.

The first assumption can be justified in that there is less of a chance of incorrectly targeting weapons if there are fewer aim points to work with. From a human error standpoint, this leads to a more reliable weapon. The second assumption is based on the method the Weapon Effectiveness Branch of the

Analysis Division, Air Force Armament Laboratory currently uses for aim point strategies (Ref 9). The assumption is useful computationally because it requires less computer time.

The search routine involves the three traits of an aim point strategy--aim point locations, number of aim points, and number of weapons per aim point. To limit the search time, a feasible range on the number of weapons is determined. Within this range, the search is conducted by varying one trait at a time. First the number of weapons per aim point is set to a common value (this weapon allocation is the aim point group designator). Next the number of aim points is fixed (this number identifies an aim point set within a group). Finally, the locations are varied until the largest PS is achieved. Then the number of aim points is increased by one, and the location search is restarted. Once this search is completed for all aim point sets within a group, the range on the number of weapons and marginal analysis are used to determine the best aim point set within this group.

Determining the Initial Bounds on the Required Number of Weapons

For a given W and V and assuming $CEP = 0$, symmetric aim point locations are determined which will satisfy the constraint in Eq (3-1). This results in a lower bound, \min , on the number of weapons. Then by relaxing the CEP assumption such that CEP equals the actual weapon's CEP and allocating one weapon to each of these locations, the probability of making the cut for a specific aim point strategy, PS, is

estimated. If $PS \geq PC$, then one has the optimal aim strategy and the subproblem is solved. If $PS < PC$, then through the use of marginal analysis and allocating only one more weapon to the current total number of weapons, an additional weapon is targeted for the aim point which increases PS the most. This weapon allocation is continued until the addition of the last weapon gives a $PS \geq PC$. By adding up the numbers of weapons targeted for each aim point in this final aim point strategy, one has an initial upper bound, max, on the required number of weapons. (See Appendix C, Subroutine BOUNDS for the computer coding of this procedure.)

Determining the "Best" Aim Point
Locations for Each Aim Point
Set Within an Aim Point Group

In order to determine the best locations for each aim point set within an aim point group, the following rules were applied: (Note Figs 5a and 5b are logic diagrams for this entire procedure.)

1. By definition, all aim points within any set will have the same number of weapons allocated to each aim point.
2. At no time will the total number of weapons allocated to an aim point set exceed the current upper bound on the number of weapons.
3. For any initial upper bound ≥ 4 , every aim point will be allocated at least two weapons.
4. For any initial upper bound ≤ 3 , every aim point will be allocated at least one weapon.

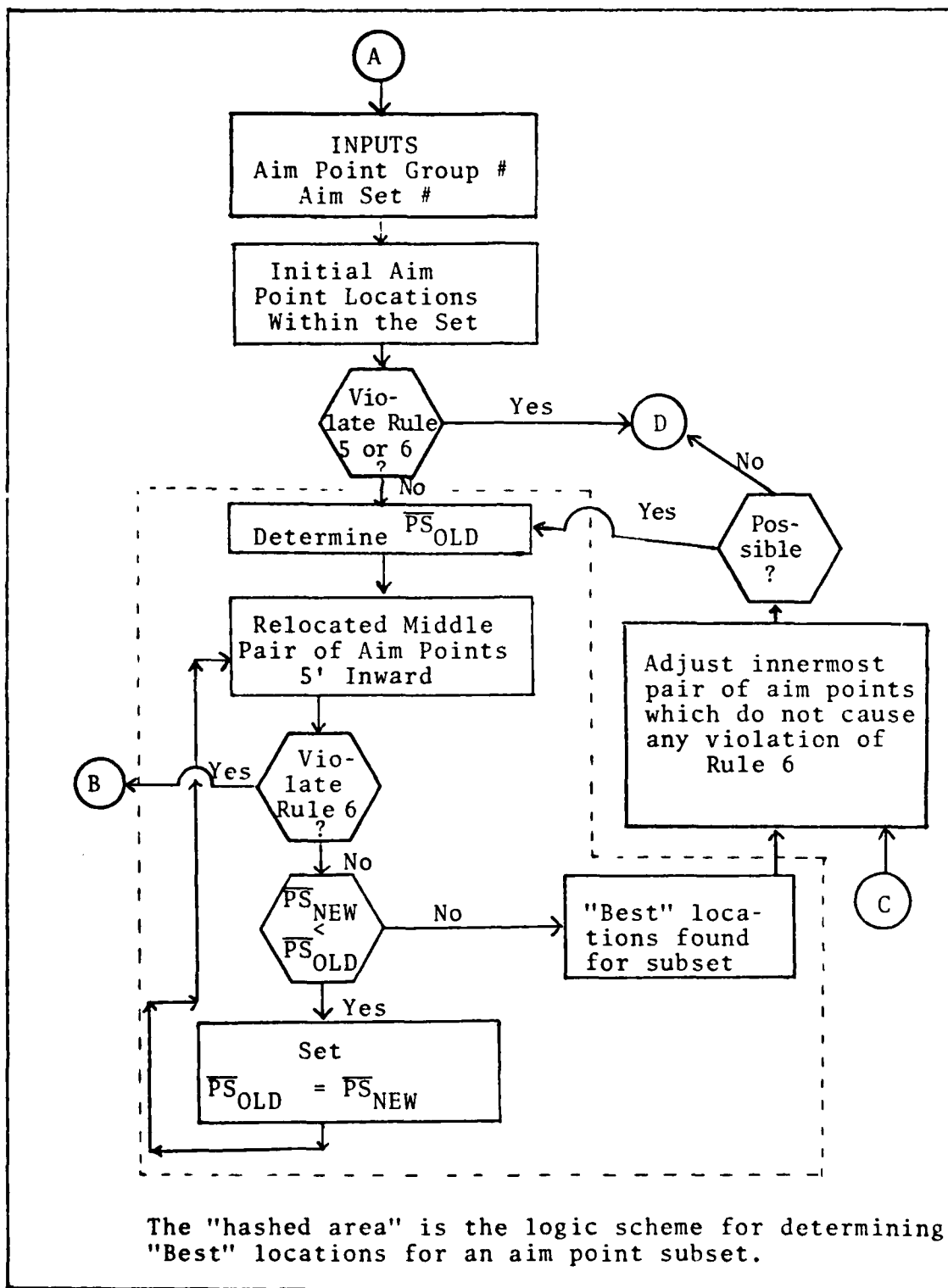


Fig 5a. Best Locations for a Set
Within a Group

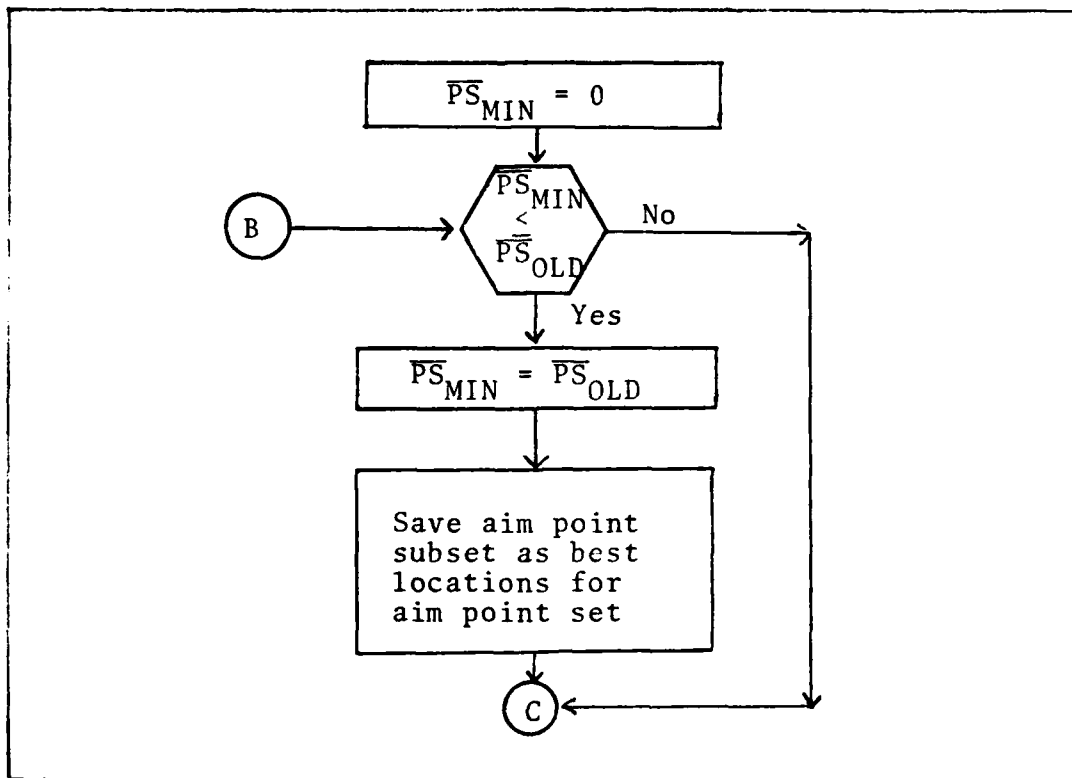


Fig 5b. Best Locations for a Set Within a Group

5. At no time will a weapon be targeted closer than one damage radius from the edge of the actual runway. This rule guarantees that at least 50 percent of all the weapons targeted for the outer pair of aim points will be 100 percent effective. By effective, it is meant that the entire crater caused by a weapon will lie within the bounds of the effective runway.

6. No two aim points will be located closer to each other than one damage radius.

7. All relocation of aim points will be in five-foot increments towards the center of the runway. Five feet was

chosen because it is ten percent of the smallest V being investigated.

To initialize the locations of the aim points within a set, each symmetric pair of aim points is located as far as possible from the center of cut without violating Rule 5 or 6. (Note: in order to maintain symmetry, the middle aim point for an odd aim point set will always be located at the center of the cut.) With these locations, \overline{PS}_{OLD} is estimated. Then the middle pair of symmetric aim points is relocated five feet inboard while holding all other aim point locations fixed. \overline{PS} is re-estimated, called \overline{PS}_{NEW} . If $\overline{PS}_{NEW} < \overline{PS}_{OLD}$, these new locations are the current "best" locations for the aim point set and \overline{PS}_{OLD} is reset to \overline{PS}_{NEW} . The middle pair is again moved five feet inboard and \overline{PS}_{NEW} is estimated again. This iterative process will stop with either of two occurrences:

1. $\overline{PS}_{NEW} > \overline{PS}_{OLD}$
2. Rule 6 is violated.

Prior to discussing the first occurrence, the following term must be defined:

An aim point subset is a specific aim point set in which all symmetric aim point pairs, except for the middle pair, are assigned specific locations. Any two subsets within an aim point set will have at least one pair of symmetric aim points, other than the middle pair, that are not co-located. Each aim point set will have at least one aim point subset; both aim point set two and three will have the null set as their only aim point subset.

With the definition of an aim point subset, one can show that \overline{PS} is a convex function for each aim point subset. Thus, as

the middle pair of aim points are moved inboard, when \overline{PS} starts to increase, the "best" locations for this aim point subset have been passed and no further searching is needed for this subset. Once the "best" locations for a specific aim point subset are found, the search algorithm moves the innermost pair of aim points (other than the middle pair), which will not cause a violation of Rule 6, five more feet towards the center of the cut. All aim points inboard of the pair just moved are initialized according to Rule 6, and this new subset is searched for its "best" location.

Each time the "best" locations for a subset are determined, the \overline{PS}_{NEW} for this aim point strategy is compared to the current minimum \overline{PS}_{OLD} for the entire aim point set. If the aim point subset's $\overline{PS}_{NEW} \leq \overline{PS}_{OLD}$, then \overline{PS}_{OLD} is reset to \overline{PS}_{NEW} , and the "best" locations for this aim point subset become the best locations for the aim point set. If $\overline{PS}_{NEW} > \overline{PS}_{OLD}$, the search continues until one has the second occurrence.

The second occurrence indicates one of two things. First, if Rule 6 is violated before $\overline{PS}_{NEW} > \overline{PS}_{OLD}$, one has too many aim points and the middle pair of aim points should be consolidated into one aim point, the center of cut. Second, if the locations of the outermost pair of aim points are such that when all inner aim points are initialized to be one damage radius apart and the middle pair of aim points violates Rule 6, the search for the best locations of aim points for this aim point set is complete.

The search algorithm then proceeds to the next aim point set within this aim point group and begins to determine the best locations for this aim point set. This whole search process stops when Rule 2 will be violated by investigating the next aim point set. (See Appendix C, Subroutine STRAT for the computer coding of this procedure.)

Determining the Best Aim
Point Set Within an Aim
Point Group

Once the best locations are determined for each aim point set (see Fig 6), the algorithm then determines which aim point set requires the fewest total number of weapons in order to have $PS \geq PC$. This is accomplished similarly to the algorithm which determined the initial upper bound. Through marginal analysis an additional weapon is targeted for the aim point within the aim point set which will cause the largest increase in PS. After each weapon is allocated, PS is compared to PC. Any of three stopping criteria are used. First, if total number of weapons allocated is equal to the current upper bound and $PS < PC$, then this aim point set is disregarded and the next aim point set is tested. Second, if $PS \geq PC$ and the total number of weapons allocated equals the lower bound on the number of weapons required, the algorithm stops and one has an optimal aim point strategy. Third, if $PS \geq PC$ and the total number of weapons allocated is less than or equal to the current upper bound, max, this aim point strategy and its PS are saved. Also, max is reset to the

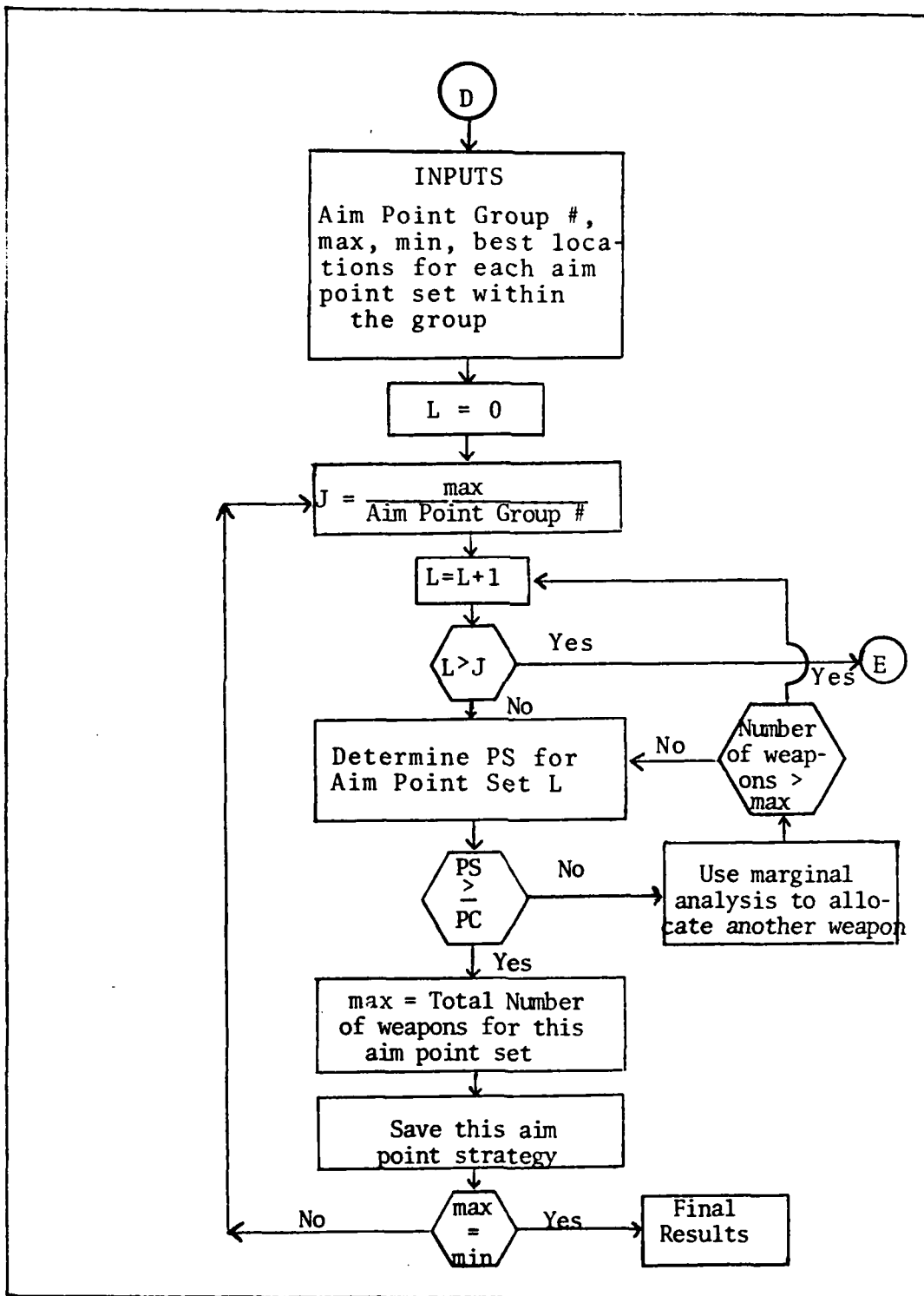


Fig 6. Best Aim Point Set Within
An Aim Point Group

number of weapons required for this aim point strategy.

Each aim point group is tested and the one requiring the fewest number of total weapons becomes the best aim point set within this aim point group. If two or more aim point sets tie as to the fewest number of weapons, the aim point set with the highest PS is then chosen as the best aim point set. (See Appendix C, Subroutine BEST for the computer coding of this procedure.)

The search algorithm then increases the number of weapons allocated per aim point by one and again starts determining the best locations for the aim point sets within this new aim point group (see Fig 7). This step is accomplished because, for example, the best locations for two aim points with two weapons per aim point may not be the best locations for two aim points with three weapons per aim point (assuming that the current upper bound is greater than or equal to six weapons). And in fact, for the following input variables:

X = 8000 feet
Y = 150 feet
W = 500 feet
U = 2000 feet
V = 50 feet
CEP = 50 feet
P = .8

the best locations for aim point group two and aim point set two are 38 feet from the leading edge of the runway and 112 feet from the leading edge of the runway. Yet with aim point group three and aim point set two, the best locations are 48 feet

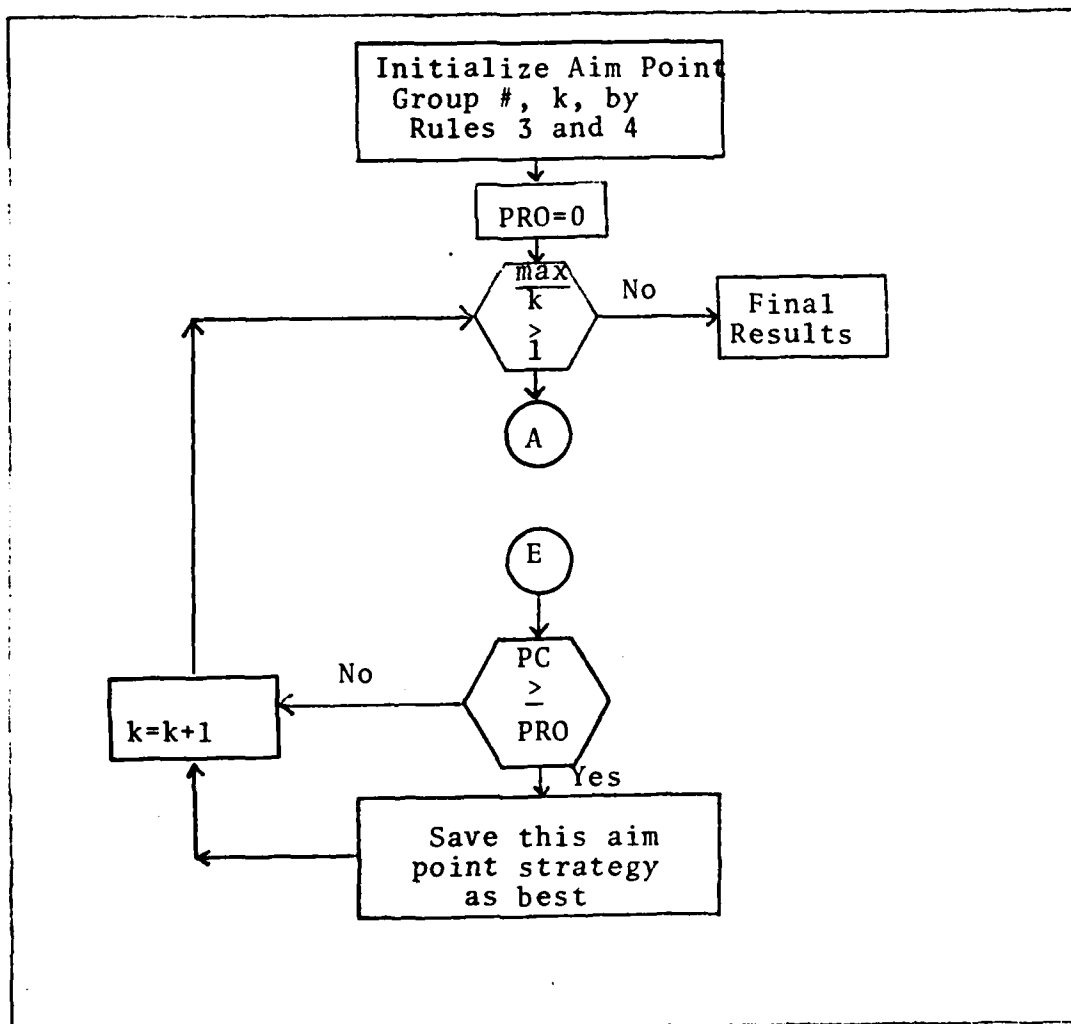


Fig 7. The Best Aim Point Strategy

from the leading edge of the runway and 101 feet from the leading edge of the runway. Thus, one cannot assume that the "best" locations for an aim point set within a specific aim point group are the best locations for that set within any group. (See Appendix C, Subroutine COUNT, for the computer coding which determines the aim point group and also stores the current best aim point strategy for all aim point groups.)

Methods for Decreasing the
Search Algorithm Calculation
Time

Two methods were investigated to decrease the search algorithm's calculation time. The first method was to use the quicker of the two \overline{PC} approximations for a given aim point strategy (either the event-composition method or the Monte Carlo simulation). The second method was to compare the probability of damaging the first takeoff width with that of the middle takeoff width for a given aim point strategy.

In using the first method to increase the efficiency (decrease the time required) of the search algorithm, the following set of input variables were used to compare the computation times for the two approximations:

W = 250 pounds
CEP = 100 feet
X = 4000 feet
U = 2000 feet
V = 100 feet
P = .8
10 aim points equally spaced from
one edge of the runway to the other
two weapons per aim point
Y = was allowed to vary

Since the driving factor, timewise, for the event-composition method is the number of L_i 's and requiring that the Monte Carlo simulation have 2770 "attacks" per estimation (the number of attacks which would provide a confidence interval of $\pm .01$ at a confidence level of .99 for a problem requiring five cuts), it was assumed that by varying the minimum launch

window's width, one could determine the numbers of L_i 's, NE for which the event-composition method was always faster than the simulation method. This data indicated that the event-composition method was always faster for $NE \leq 12$. Even though this rule ($NE \leq 12$; implies event-composition method will be faster) is not always true, this fact was not known until the model was verified.

As to the second method for a more efficient search algorithm, given the "best" locations within an aim point subset, the probability of damaging the first minimum launch window and the middle minimum launch window were assumed to be approximately equal to each other. Thus, by comparing these two probabilities, one could reject some aim point strategies from consideration without having to use either method for approximating \overline{PC} . The author chose to use a difference $\leq .05$ as being approximately equal and thus worthy of further investigation. (See Appendix C, Subroutine APROX for the computer coding of this procedure.)

Summary

To use either of the two approximating methods for \overline{PC} , one must first determine an aim point strategy. This chapter discussed a search routine which not only determines aim point strategies, but also solves for the aim point strategy which minimizes the total number of weapons required subject to the inherent limitations of this search algorithm. Finally, two methods for increasing the efficiency of the search routine were discussed.

V. Verification and Validation of the Model

Introduction

Coupling the two methods for approximating \overline{PC} with the search algorithm, one has a model which will determine a good aim point strategy. Yet this model is of questionable value until it has been verified and validated. Fishman and Kiviat (Ref 6) define verification and validation as:

Verification - insuring that the model behaves the way
the experimenter intends

Validation - testing the agreement between the be-
havior of the model and that of the real
system.

Shannon further expands on validation by stating, "Validation is the process of bringing to an acceptable level the user's confidence that any inference about a system derived from the simulation (model) is correct [Ref 12: 29]." Furthermore, he writes

The concept of validation should be considered one of degree and not one of an either-or notion; it is not a binary decision variable where the model is valid or invalid. It is not at all certain that it is ever theoretically possible to establish if we have an absolutely valid model; even if we could, few managers would be willing to pay the price [Ref 12: 208].

The procedure used to verify and validate the model was to first verify if the model behaves as intended, secondly

to verify the part of the model which estimates PC, and thirdly to "validate" the model by comparing its results to the Airfield Assessment Program, AAP (formerly called "MASSIVE") used by the Weapons Effectiveness Branch, Analysis Division, Air Force Armament Laboratory, Eglin AFB, Florida (Ref 3).

Verification

All the subroutines within the model were verified through the use of print statements and hand calculations to determine whether they "behave" in the manner desired. Yet, the overall results of the model, the recommended aim point strategies, are extremely dependent upon the accuracy of the two approximation methods for \overline{PC} . Thus, these two approximation methods received a more in-depth verification than the rest of the model.

Verifying the Two Approximating Methods for \overline{PC} . After the FORTRAN coding for the event-composition method and the Monte Carlo simulation method was shown to do what it was supposed to do, the designer checked the accuracy of both methods. Since the event-composition method is a discrete approximation of a continuous process, a discrete probability situation was used in its initial verification.

Figures 8a, b, and c show the cases tested against the event-composition approximation. In each case, the actual value of $P(\overline{E})$, the probability of making the cut, was calculated manually and $P(E)$, the probability of not making the cut, was calculated using the event-composition method. The results in

Figs 8a, b, and c indicate confidence in this calculation of $P(E)$. One should note the following prior to reviewing the figures:

1. The values between the hash marks are the probabilities that a weapon targeted for an aim point will impact between these hash marks.
2. The number of hash lines is indicative of the number of aim points (A.P.).
3. If two weapons are targeted for the same point, each has the impact probabilities shown.

Once this simplified event-composition approximation was verified, this method was used to verify the Monte Carlo simulation approximation. A subproblem was designed with each event L_i composed of exactly two event A_{jki} , where the length over which an A_{jki} might occur equals the standard deviation of the weapon being employed. Along with these specifications, the following input variables were used:

$R = 10$ feet

$Y = 70$ feet

$V = 20$ feet

Two aim points, each located 5 feet either side of the center of the runway's width

One weapon targeted for each aim point

The $P(A_{jki})$'s were calculated using the formulation discussed in Chapter III. Both the event-composition method and the Monte Carlo simulation method were given these inputs, and following results were obtained:

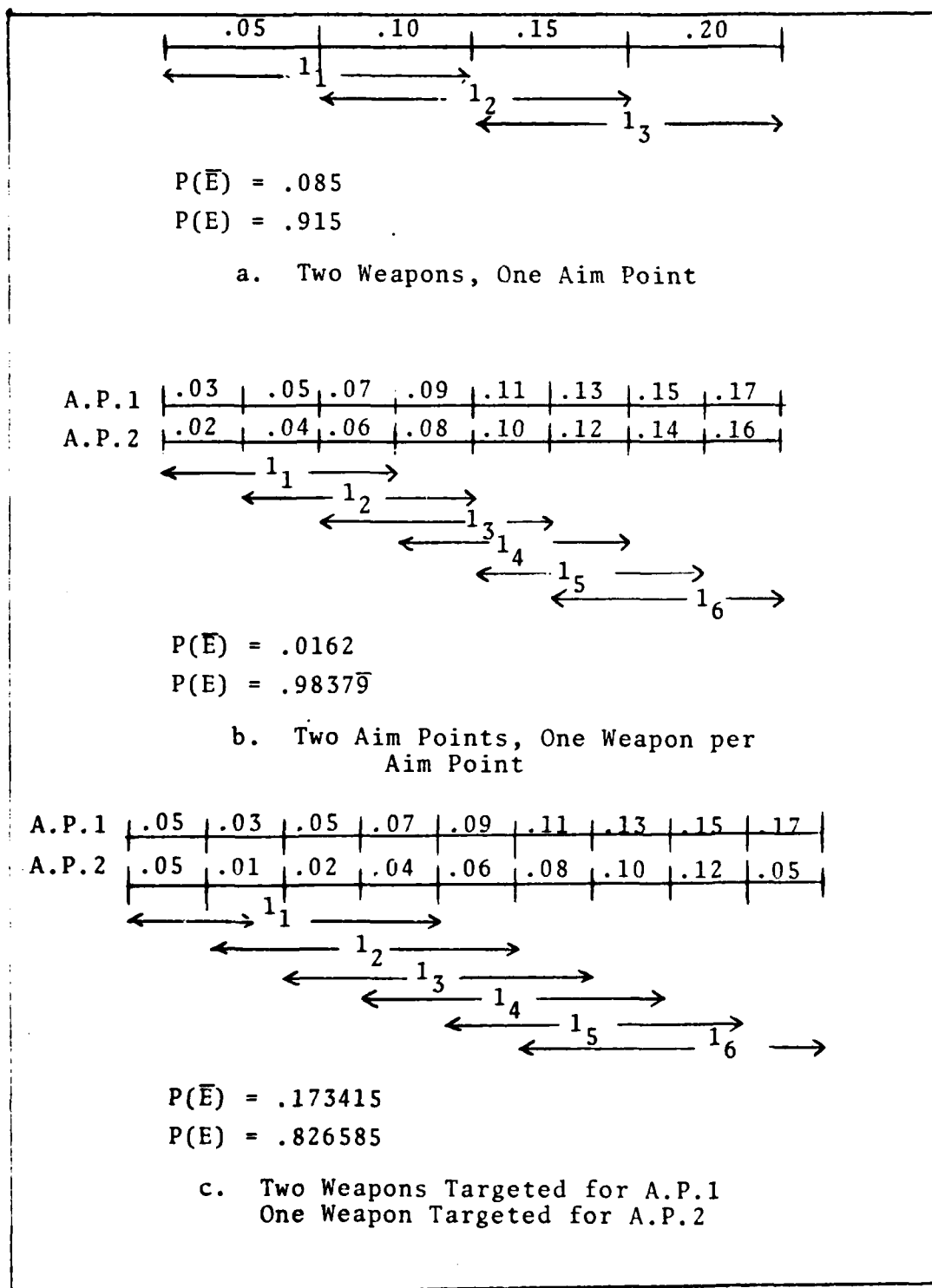


Fig 8. Comparison of $P(\bar{E})$ to $P(E)$

$P(E)$ for the event-composition = .732519660803

$P(E)$ for the simulation = $P(.7206 \leq x \leq .7434) = .99$
with 10,000 "attacks"; the point estimation for
the simulation's $P(E) = .732$

This seemed to verify the simulation, but one more check was run. This next verification step used the following input data:

W = 250 pounds
CEP = 50 feet
X = 8000 feet
Y = 150 feet
U = 2000 feet
V = 100 feet
P = .8

The entire model was run 31 times using a uniform random number generator to provide a different "seed" for the normal random number generator on each run. This verification step was taken in order to demonstrate consistency in the simulation method. The following results were obtained:

the mean of the 31 observations, $\bar{X} = .960429$
the unbiased sample variance, $S = .00505$

Assuming the Central Limit Theorem applied to these results (Ref 11: 252), one would expect about 21 observations within one standard deviation of \bar{X} and about 29 observations within two standard deviations of \bar{X} . The actual results had 25 observations within the first interval and all 31 observations within the second interval. Due to the smallness of the number of observations taken, these discrepancies were not considered severe, and the simulation method was assumed to be consistent and not "seed" dependent.

With the verification of the simulation method completed, a number of test cases were run to compare the simulation method and the event-composition method. (Note: when the simulation method was verified using the event-composition method, the subproblem was constructed using a very simplified event-composition. That simplification was not based on the same size Z's described in Chapter III. However, this comparison will actually compare the simulation and the event-composition methods as developed in Chapter III.) All input variables were held constant except for W, CEP and Y. Furthermore, in each case there were four aim points located at .2Y, .4Y, .6Y and .8Y; and one weapon was targeted for each aim point (see Table I).

TABLE I
Comparison of \overline{PC} Approximation Methods

Inputs			\overline{PC} Approximations	
W	CEP	Y	Event-Composition	Simulation (.95 confidence interval)
1000	20	300	.000484	.0 to .061980
1000	100	150	.051687	.061378 to .094621
1000	100	300	.388422	.411219 to .472781
250	20	300	.001631	.0031465 to .014853
X = 4000, U = 2000, V = 100				

The results of these comparisons indicated:

1. Though the event-composition method was designed to be "optimistic", i.e. give a larger \overline{PC} than the actual attack, this approximation seems to be "pessimistic."
2. Within the range of PC under investigation, .946

to .997 ($P = .8$; CEP's such that one has four or five cuts), the event-composition method would probably not be enough in error to suggest using fewer weapons than would actually be required. (This assumption was shown to be true for the range of input variables investigated. See Table II)

The conclusion derived from these results was to use the event-composition method in the search algorithm when it provided a time savings, but to always use the Monte Carlo simulation method to approximate the resulting PC from the best aim point strategy. This procedure would provide a built-in model verification for any aim point strategy determined "best" by the event-composition method.

Verification of the Time Saving Methods. As was indicated in Chapter IV, using 12 L_1 's as the changeover point between the efficiency, timewise, of the event-composition method and Monte Carlo simulation turned out to be in error. Using the following set of input data:

W = 1000 pounds
CEP = 20 feet
X = 8000 feet
Y = 150 feet
U = 2000 feet
V = 60 feet
P = .8
3 Symmetric Aim Points
2 Weapons per Aim Point
3470 "Attacks" for the simulation

the simulation method was ten times faster than the event-composition method. Due to lack of time, the relationship

between the number of weapons (the major determinant of the time for the simulation) and the number of L_i 's (the driving factor, timewise, for the event-composition method) was not investigated further.

The second efficiency method used in the search routine, setting a tolerance on the difference between the probability of destroying the first V and the probability of destroying the middle V, was verified with the following data:

W = 500 pounds
CEP = 50 feet
X = 8000 feet
Y = 150 feet
U = 2000 feet
V = 50 feet
P = .8

The above data was run with two tolerances, the actual tolerance used in the model, .05, and the tolerance of 1.0. Both of these computer runs determine the same best aim point strategy, yet the run with a tolerance of .05 was 1.4 seconds (approximately one percent) quicker.

Factorial Design. Due to the lack of time, only a 2^3 factorial design was constructed to investigate the model. The following input variables were held constant for each run of the model:

X = 8000 feet
Y = 150 feet
U = 2000 feet
P = .8

The remaining input variables were set at the following levels:

	<u>High</u>	<u>Low</u>
W	1000 pounds	250 pounds
CEP	100 feet	20 feet
V	100 feet	50 feet

Table II provides the levels of the varied inputs and the model's results for each run. Using the number of weapons required to make the cut as the dependent variable and assigning qualitative value (zero for low and one for high) to each of the three independent variables, an analysis of variance (ANOVA), was performed "... to locate important independent variables in a study and determine how they interact and affect the response (the dependent variable) [Ref 11:458]." In order to accomplish the ANOVA, the three-way interaction of the independent variables had to be suppressed. This suppression was caused by too little data to investigate the main effects and all their interaction. The result of this ANOVA suggested the only significant effects on the number of weapons required were caused by W, CEP, and V and not their interactions. Thus, a second ANOVA was accomplished suppressing all interactions. The result of this ANOVA are given in Table III. Table III indicates, for a Type I error (Ref 11:328) of less than .05, the only independent variables (of those investigated) which definitely affect the number of weapons required are CEP and V.

With the insight provided by the ANOVA in Table III, a second type of analysis was attempted. This analysis was a linear regression using the same dependent and independent

TABLE II

Results* from the 2^3 Factorial Design

INPUTS			RESULTS			
W	CEP	V	Aim Point Locations	Weapons Per Aim Point	$P(a \leq PC \leq b) = .99$	Cut Locations**
1000	100	50	75	8	.954982 to .974982	1739,3223, 4708,6193, 7678 (1)
250	20	50	42,107	2,2	.981485 to 1.0	1943,3835, 5727,7620 (2)
250	100	100	75	5	.951371 to .971371	(1)
1000	20	50	35,75,114	1,1,1	.989119 to 1.0	(2)
250	20	100	75	1	.984715 to 1.0	(2)
250	100	50	37,75,115	3,5,3	.948483 to .968483	(1)
1000	20	100	75	1	.989706 to 1.0	(2)
1000	100	100	75	4	.957870 to .977870	(1)

*The following inputs were held constant: $X=8000$; $Y=150$; $U=2000$; and $P=.8$ **Due to the derivation of the subproblem, the only factor changing cut locations for these sets of data was CEP. Thus all runs with CEP=100 feet had cut location group one, (1), and those with CEP=50 feet had group two, (2). For the derivation \bar{P} , the estimated probability of destroying the runway, see Appendix D.

+These values are largely due to Rule 3 and assumption that all cuts will be attacked with like numbers of weapons.

TABLE III
ANOVA of the Main Effects of W, CEP,
and V on the Number of Weapons
Required to Make a Cut

Source of Variation	Sum of Squares	Degrees of Freedom	Mean Square Error	F	Significance of F
W	3.125	1	3.125	2.273	.206
CEP	45.125	1	45.125	32.818	.005
V	28.125	1	28.125	20.455	.011
Residual	5.500	4	1.375		
Total	81.875	7	11.696		
with X = 8000, Y = 150, U = 2000, and P = .8					

variable. Yet, the independent variables were now treated as quantitative factors rather than qualitative, as in the ANOVA. The following linear equation was obtained:

$$\begin{aligned} \text{the number of weapons required} = \\ .059375\text{CEP} - .075\text{V} + 6.6875 \end{aligned} \quad (5-1)$$

The coefficients of CEP and V and the constant term are all significant for a Type I error $\leq .01$ (that is, one is 98 percent confident that none of these values are actually zero), and for the adjusted R^2 value 89.5 percent of the variance in the number of weapons required is explained by Eq (5-1).

"Validation" of the Model

As was pointed out in the introduction to this chapter, it may very well be impossible to establish a model as absolutely valid (Ref 12: 208). Yet, Shannon states, "... it is

the operational utility of the model and not the truth of its structure that usually concerns us [Ref 12: 29]." It was felt that the best measure of the "operational utility" of this model would be to compare its results with a model such as AAP which is currently being used by the Air Force for similar types of investigation (Ref 3).

Before discussing the comparison, one should understand that AAP was designed to determine the probability of destroying a runway given the aim point strategy. During conversations with Mr. Bass (Ref 3), it became evident that to obtain similar results, he would need to run a number of different aim point strategies. Plus, the aim point strategies he employed would have to be based on his intuition gained from dealing with similar problems in this field. (Note: for an in-depth discussion of AAP, see Ref 2.)

As to the comparison of the results of the two models, the following input data were used:

W = 500 pounds
CEP = 50 feet
X = 8000 feet
Y = 150 feet
U = 2000 feet
V = 50 feet
P = .8

Mr. Bass' "best" aim point strategy was:

1. Total # of weapons was 24
2. Three weapons were on each of two aim points
3. These aim points were located 50 feet from one

edge of the runway and 50 feet from the other edge.

4. Four cuts were required to be made in the runway.
5. The cuts were located at 1600-foot intervals along the runway (1600, 3200, 4800, 6400 feet from the end of the runway).
6. The estimated probability of destroying the runway was .88.

This model obtained the following results:

1. Total # of weapons was 24
2. Three weapons on each of two aim points
3. These aim points were located 48 feet from one edge of the runway and 49 feet from the other edge (lack of symmetry is due to rounding the aim point locations off to whole numbers of feet)
4. Four cuts were required in the runway
5. The cuts were located at a distance of 1865 feet, 3605 feet, 5345 feet, and 7084 feet from the end of the runway
6. The point estimate for the probability of destroying the runway was .908

This aim point strategy was then employed using AAP, and the point estimate for the probability of destroying the runway increased to .89 with a standard deviation of .314 (only 100 "attacks" were conducted).

Despite the fact that only one comparison was made between the model developed in this thesis and AAP, it seems clear that there is some "operational utility" in the model.

This comparison showed this model determined a "better" aim point strategy than the one derived by Mr. Bass' intuition and AAP. Yet, one comparison cannot be used as validation for this model. What can be ascertained is that, for the case investigated, the results of the model are not invalid. Further cases should be tested before one could suggest that the model is valid, but due to time constraints on the AFIT computer and Mr. Bass' schedule, the investigation was not continued.

Summary

This chapter discussed the verification and validation of the model. The steps taken in verifying the model were 1) making sure that each of the subroutines accomplished what the designer intended it to; 2) an in-depth verification of both methods of approximating \overline{PC} , given an aim point strategy; and 3) using an ANOVA and linear regression to determine the effects of W, CEP and V on the number of weapons required to achieve PC. The "validation" consisted of comparing the results of this model to those of AAP for a specific set of input variables.

VI. Conclusions and Recommendations

This study reports the development of a model which determines an aim point strategy for reducing the total number of weapons employed to achieve a desired level of runway destruction. Initially, the overall problem is reduced to a number of identical, stochastically independent subproblems. Using one of these subproblems, two methods for estimating its destruction criteria are developed. Then, these estimating methods are coupled with a search algorithm to determine the "best" aim point strategy. Finally, the entire model is verified and "validated." From the effort expended on this study, a number of conclusions and recommendations can be made.

Conclusions

1. The model, when compared to the current Air Force work in this area, seems to provide aim point strategies which are better than those currently used.
2. The results of this study indicate an area of possible savings for the Air Force. This savings is brought about by reducing the number of weapons employed while maintaining the desired level of destruction.
3. This study, in its entirety, provides a basis for future investigation on targeting conventional weapons against non-reinforced concrete runways.

Recommendations

1. Since symmetric aim points are currently acceptable in determining an aim point strategy (Ref 3), the approximation methods could be changed to incorporate this type of strategy and, thus, further increase their efficiency timewise.

2. Further comparisons between AAP and this model need to be performed to increase the user's confidence level in the model's results.

3. One could adapt the model to investigate CEPs which cannot logically be described by circular normal probability distributions.

4. One could adapt the model to use different aim point strategies for some cuts and possibly further reduce the total number of weapons required to achieve the runway destruction criterion.

5. Due to the availability of data, this model determines aim point strategies for destroying non-reinforced concrete runways. A slight modification in Subroutine CONVRT would allow investigations of reinforced concrete runways which are more typical of current operational runways.

Bibliography

1. Alexander, Edward T., Jr., et al. "Conceptual Design and Analysis of a Boost-lift Stand-off Missile." (U) Secret. Unpublished MS thesis, School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, December 1979.
2. Ballog, Dennis F., and Darrell B. Hutchinson. "The Determination of the Accessibility of Post-Attack Launch Windows." Unpublished MS thesis, School of Systems and Logistics, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, August 1979. AD A076925.
3. Bass, Jerry. Weapon Effectiveness Analyst, Weapons Effectiveness Branch, Analysis Division, Air Force Armament Laboratory, AFATL/DLYW, Eglin AFB, Florida. Telephone interviews, 1-10 November 1980.
4. Breuer, D. Wallace. Professor of Aeronautical Engineering. "Probability and Statistics in Weapon Analysis." Course Notes, MC 5.80, Fundamentals of Weapons Systems. Air Force Institute of Technology, Wright-Patterson AFB, Ohio.
5. Chabai, J. "On Scaling Dimension of Craters Produced by Buried Explosives," Journal of Geophysical Research, 70:5075 (1965).
6. Fishman, G.S., and P.J. Kiviat. "The Analysis of Simulation-Generated Time Series," Management Science, 13: 7 (March 1967).
7. Hokanson, Lawrence D. "Airfield Bomb Damage Repair," The Military Engineer, September-October 1975, pp. 260-62.
8. Jane's All the World's Aircraft 1978-1979. New York: Jane's Yearbooks, Franklin Watts, Inc., 1978.
9. JMEM/BASIC. Joint Munition Effectiveness Manual: Air-to-Surface Weapons Effectiveness, Selection and Requirements. (U) Secret. Washington DC: Department of Defense.
10. JMEM/WEAPONS CHARACTERISTICS. Joint Munitions Effectiveness Manual: Air-to-Surface Weapons Characteristics (U) Secret. Washington DC: Department of Defense.

11. Mendenhall, Williams, and Richard L. Scheaffer.
Mathematical Statistics With Applications. North
Scituate, MA: Duxbury Press, 1973.
12. Shannon, Robert E. Systems Simulation: The Art and
Science. Englewood Cliffs, NJ: Prentice-Hall, Inc.,
1975.
13. Waterways Experiment Station. Waterways Experiment
Station Technical Report, Number N-79-2.

APPENDIX A

Derivation of the Effective Explosive

Weight Conversion Formula

Derivation of the Effective Explosive

Weight Conversion Formula

For a known effective explosive weight, W, the optimal DOB can be determined from the following inequality (Ref 10: 1-133):

$$1 < \text{DOB}/W^{1/3} < 1.5 \quad (\text{A-1})$$

for $W > 50$ pounds of TNT. Assuming the midpoint of this interval (1.25) to be a good approximation of this ratio, one can then determine the scaled radius ($\text{ft}/\text{lb}^{1/3}$) by referring to Figs 5.9 and 5.10 in Waterways Experiment Station Technical Report Number N-76-2 (Ref 13:100-01). For a W greater than 50 pounds of TNT detonating in dry-to-moist clay beneath a non-reinforced concrete runway (Ref 13:100), the scaled radius is approximately $3.65 \text{ ft}/\text{lb}^{1/3}$. For a W greater than 50 pounds of TNT detonating in dry-to-moist sand beneath a non-reinforced concrete runway (Ref 13:101), the scaled radius is approximately $3.54 \text{ ft}/\text{lb}^{1/3}$.

These scaling radii are derived from a scaling factor determined by Chabai (Ref 4):

$$\frac{R_1}{R_2} = \left(\frac{\rho_1}{\rho_2} \right)^{1/3} \left(\frac{W_1}{W_2} \right)^{1/3} \quad (\text{A-2})$$

where R is the damage radius, and ρ is the density of the

material in which the explosion occurs. The subscript 1 refers to a set of known data. For this use $\rho_1/\rho_2 = 1$, thus

$$\frac{R_1}{R_2} = \left(\frac{W_1}{W_2}\right)^{1/3} \quad (A-3)$$

and

$$\frac{R_1}{(W_1)^{1/3}} = \frac{R_2}{(W_2)^{1/3}} \quad (A-4)$$

The expression on the righthand side of Eq (A-4) is the scaled radius. Since the density ratio (ρ_1/ρ_2) is assumed to be one, the scaled radius will be assumed to be the lesser of the two from the previous paragraph ($3.54 \text{ ft/lb}^{1/3}$). Thus

$$R = (3.54)(W)^{1/3} \quad (A-5)$$

For example, with an effective explosive weight of 1000 pounds TNT, the damage radius would be 35.4 feet.

APPENDIX B

Derivation of the Circular Error

Probable Conversion Formula

Derivation of the Circular Error

Probable Conversion Formula

For a bivariate normal probability distribution, where the x and y variables are independent, the joint density function is

$$f(x,y) = \left\{ \left(\frac{1}{\sigma_x \sqrt{2\pi}} \right) \exp - \frac{1}{2} \left(\frac{x-\mu}{\sigma_x} \right)^2 \right\} \left\{ \left(\frac{1}{\sigma_y \sqrt{2\pi}} \right) \exp - \frac{1}{2} \left(\frac{y-\mu}{\sigma_y} \right)^2 \right\} \quad (B-1)$$

The first assumption in Chapter I is that the CEP of the weapon can logically be described by the circular normal distribution. This implies that the standard deviations of the joint distribution are ". . . independent of direction or choice of coordinate axis. . ." (Ref 4:14), or more specifically, $\sigma_x = \sigma_y$. Thus, applying this assumption to Eq (B-1), one obtains

$$f(x,y) = \frac{1}{2\pi\sigma^2} \left\{ \exp - \frac{1}{2} \left(\frac{x-\mu}{\sigma} \right)^2 \right\} \left\{ \exp - \frac{1}{2} \left(\frac{y-\mu}{\sigma} \right)^2 \right\} \quad (B-2)$$

A further simplification is achieved by realizing that μ is actually the aim point of the weapon and, thus, $\mu = 0$.

$$f(x,y) = \frac{1}{2\pi\sigma^2} \exp - \frac{1}{2} \left(\frac{x^2+y^2}{\sigma^2} \right) \quad (B-3)$$

Letting $r^2 = x^2 + y^2$, Eq (B-3) becomes (Ref 4:14):

$$f(r) = \frac{1}{2\pi\sigma^2} \exp - \frac{1}{2} \left(\frac{r}{\sigma} \right)^2 \quad (B-4)$$

Yet CEP is defined such that

$$.5 = \int_0^{\text{CEP}} f(r) dr \quad (\text{B-5})$$

Thus,

$$.5 = \int_0^{\text{CEP}} \frac{1}{2\pi\sigma^2} \exp - \frac{1}{2}\left(\frac{r}{\sigma}\right)^2 dr \quad (\text{B-6})$$

$$.5 = 1 - \exp - \frac{1}{2}\left(\frac{\text{CEP}}{\sigma}\right)^2 \quad (\text{B-7})$$

$$2 = \exp - \frac{1}{2}\left(\frac{\text{CEP}}{\sigma}\right)^2 \quad (\text{B-8})$$

$$\ln 2 = \frac{1}{2}\left(\frac{\text{CEP}}{\sigma}\right)^2 \quad (\text{B-9})$$

$$2 \ln 2 = \left(\frac{\text{CEP}}{\sigma}\right)^2 \quad (\text{B-10})$$

$$\sigma = \frac{\text{CEP}}{(2 \ln 2)^{1/2}} \quad (\text{B-11})$$

APPENDIX C
FORTTRAN Coding for the Model

FORTTRAN Coding for the Model

PROGRAM MAIN(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)

C*****

C THIS PROGRAM IS DESIGNED AROUND A MODULAR APPROACH
C MOST OF THE CALCULATIONS ARE DONE IN THE SUBROUTINES
C THE FOLLOWING VARIABLES ARE:

C 1. W=THE EFFECTIVE EXPLOSIVE WEIGHT OF THE WEAPON
C 2. CEP=THE CIRCULAR ERROR PROBABLE OF THE WEAPON
C 3. RUNLEN=THE ACTUAL RUNWAY LENGTH
C 4. RUNWID=THE ACTUAL RUNWAY WIDTH
C 5. TOLEN=THE MINIMUM TAKEOFF LENGTH
C 6. TOWID=THE MINIMUM TAKEOFF WIDTH
C 7. PRORUN=THE RUNWAY DESTRUCTION CRITERIA'S CONFIDENCE LEVEL

C*****

DIMENSION MINAP(20),MINBMB(20)
DIMENSION AIMLOC(20),NUMBOM(20),SECBOM(20,20),PROSEC(20,20)
DIMENSION PROFAC(20,20),DAMLOC(20)
DIMENSION INDX(20),T(20),SUM(20),SECFRA(20,20)
DIMENSION LOCCUT(20),APMIN(20),MINBOM(20),BESTAP(20,20),TMRGIN(20)
DIMENSION VALUE(20),AIMGP(20,20)
DIMENSION BVALUE(20)
DIMENSION HIT(50),R1(1)
DIMENSION IND(20)
DIMENSION EQUIPRO(2,20,25),S1(25),PROD(2),PROEQU(2,20,25)
COMMON PROSPG(12,20,20)
COMMON/JOHN/SECBOM,PROSEC,PROFAC,DAMLOC,SECFRA
COMMON/CUT/LOCCUT
COMMON/MARGIN/TMRGIN
COMMON/AIMGRP/AIMGP
COMMON/INDICE/INDX
COMMON/FIXED/RUNWID,DAMRAD,STADEV,TOWID,PROCUT
COMMON/OPT/BESTAP,IND,BVALUE
COMMON/SIMS/HIT,R1
COMMON/ITERAT/NITERA
COMMON/GRP/NSPG,NSPW,NGPW,STEP,FRAI,ERUNWI,ERWID
COMMON/EQUAL/MINAP,MINNBR,MINBMB,NUMBER
COMMON/APPROX/EQUIPRO,S1,PROD,PROEQU

C*****

500 FORMAT(1H1)

RANSET=585285124.

100 READ*,W,CEP,RUNLEN,RUNWID,TOLEN,TOWID,PRORUN
IF(EOF(5).NE.0) GO TO 20

C INPUTS TO THE SYSTEM

WRITE(6,500)
PRINT*," "
PRINT*," THE EFFECTIVE EXPLOSIVE WEIGHT OF THE WEAPON IS ",W," POU
CNDS OF TNT."
PRINT*," "
PRINT*," THE CEP OF THE WEAPON IS ",CEP," FEET."
PRINT*," "
PRINT*," THE ACTUAL RUNWAY DIMENSIONS ARE ",RUNLEN," FEET BY ",R
CUNWID," FEET."
PRINT*," "
PRINT*," THE MINIMUM LAUNCH WINDOW DIMENSIONS ARE ",TOLEN," FEET
CBY ",TOWID," FEET."
PRINT*," "
PRINT*," THE RUNWAY DESTRUCTION CRITERIA IS ",PRORUN," ."
PRINT*," "
PRINT*," "

C THE CALLS TO THE SUBROUTINES WHERE MOST OF THE WORK IS DONE

CALL CONVRT(W,CEP,RUNLEN,TOLEN,PRORUN,NCUT)
CALL GROUP
CALL ITER
CALL BOUNDS(MIN,MAX,NUMAIM)
IF(MAX.EQ.MIN) NUMBER=NUMAIM
IF(MAX.EQ.MIN) GO TO 15
ICOUNT=1
MINN3R=MIN
MINNUM=MAX
IF((MAX.GE.3).AND.(MAX.LE.4)) ICOUNT=0
CALL COUNT(ICOUNT,MINNUM)

C THE PRINT STATEMENTS FOR THE OUTPUT

```

15  CONTINUE
    PRINT*, " "
    MINNBR=0
    DO 21 I=1,NUMBER
      PRINT*, " ", MINBMB(I), " WEAPONS SHOULD BE TARGETED FOR ", MINAP(I) ,
C" FEET FROM THE EDGE OF THE RUNWAY."
      AIMLOC(I)=MINAP(I)
      MINNBR=MINNBR+MINBMB(I)
21  CONTINUE
    PRINT*, " "
    DO 22 I=1,NCUT
      PRINT*, " THE LOCATION OF CUT NUMBER ", I, " IS ", LOCCUT(I), " FEET FR
C" OM THE LEADING EDGE OF THE RUNWAY."
22  CONTINUE
    NTOTAL=NCUT*MINNBR
    PRINT*, " "
    CALL SIM(NUMBER, AIMLOC, MINBMB, TOTAL)
    Y=1.-TOTAL
    Y1=Y-.01
    Y2=Y+.01
    IF(Y2.GT.1.) Y2=1.
    PRINT*, " WITH THIS AIM POINT STRATEGY, ONE IS 99 PERCENT CONFIDENT
C THAT THE ACTUAL CUT DESTRUCTION CRITERIA IS BETWEEN ", Y1, " AND ",
CY2
    PRINT*, " "
    PRINT*, " THE TOTAL NUMBER OF WEAPONS REQUIRED IS ", NTOTAL
    PRINT*, " "
    X1=NCUT
    CALL RUNPRO(Y, NCUT, BNDLOW, BNDUP)
    IF(BNDUP.GT.1.) BNDUP=1.
    PRINT*, " WITH THIS AIM POINT STRATEGY, ONE IS 99 PERCENT CONFIDENT
C THAT THE ACTUAL RUNWAY DESTRUCTION CRITERIA IS BETWEEN", BNDLOW,
C" AND ", BNDUP
1000 CONTINUE

```

C*****

```

20  GO TO 100
    STOP "END OF PROGRAM"
    END

```

Subroutine CONVRT

SUBROUTINE CONVRT(W,CEP,RUNLEN,TOLEN,PRORUN,NCUT)

C THIS SUBROUTINE (CONVRT) DOES THE INITIAL CONVERSION ON THE INPUT
C VARIABLES. IT RETURNS TO THE MAIN PROGRAM ALL THE INFORMATION
C REQUIRED TO CONTINUE PLUS THE LOCATIONS OF THE CUTS ARE ROUNDED
C OFF TO THE NEAREST INTEGER NUMBER OF FEET AND STORED IN LOCCUT
C THE FOLLOWING VARIABLES ARE:

- C 1. DAMRAD=THE DAMAGE RADIUS
- C 2. STADEV=THE STANDARD DEVIATION
- C 3. TOM=MINIMUM TAKEOFF LENGTH FOR ACROSS THE RUNWAY TAKEOFFS
- C 4. NCUT=THE NUMBER OF CUTS REQUIRED IN THE RUNWAY
- C 5. PROCUT=THE REQUIRED PROBABILITY FOR ANY ONE CUT
- C 6. LOCCUT=AN ARRAY OF THE LOCATIONS OF THE CUTS

DIMENSION LOCCUT(20)
COMMON/CUT/LOCCUT
COMMON/FIXED/RUNWID,DAMRAD,STADEV,TOWID,PROCUT
COMMON/GRP/NSPG,NSPW,NGPW,STEP,FRAI,ERUNWI,ERWID

DAMRAD=3.54*((W)**(1./3.))
STADEV=(CEP)/((2.*ALOG(2.))**(1./2.))
ERUNWI=RUNWID+2.*DAMRAD
TOM=((TOLEN)**2.-(RUNWID)**2.)*(1./2.)
BCI=TOM-6.*STADEV
NCUT=(RUNLEN-(TOM-3.*STADEV))/BCI+1
X=NCUT
PROCUT=(PRORUN)**(1./X)
LOCCUT(1)=TOM-3.*STADEV
Y=LOCCUT(1)
DO 1 I=2,NCUT
Z=I-1
LOCCUT(I)=Y+Z*BCI
1 CONTINUE
RETURN
END

Subroutine GROUP

SUBROUTINE GROUP

C*****

C THIS SUBROUTINE (GROUP) IS DESIGNED TO PROVIDE
C THE NECESSARY INFORMATION FOR THE EVENT COMPO-
C SITION METHOD AND ALSO TO COMPUTE THE NUMBER
C OF GROUPS PER WIDTH (NGPW).

C THE FOLLOWING VARIABLES ARE:

- C 1. STEP=THE SIZE OF THE SECTION
- C 2. ERUNWI=THE EFFECTIVE RUNWAY WIDTH
- C 3. ERWID=THE EFFECTIVE RUNWAY WIDTH MINUS THE
C LAST DAMAGE RADIUS
- C 4. NSPW=NUMBER OF SECTION PER WIDTH
- C 5. NSPG=NUMBER OF SECTIONS PER GROUP
- C 6. FRAC=THE FRACTION OF TOWID WHEN COMPARED TO
C THE DAMAGE RADIUS
- C 7. FRA=THE FRACTION OF THE GROUP WHEN COMPARED
C TO STEP
- C 8. NGPW=NUMBER OF GROUPS PER WIDTH

C*****

COMMON/FIXED/RUNWID,DAMRAD,STADEV,TOWID,PROCUT
COMMON/GRP/NSPG,NSPW,NGPW,STEP,FRAI,ERUNWI,ERWID

C*****

A85=.1
STEP=TOWID*A85
FRAC=TOWID/DAMRAD
NFRAC=TOWID/DAMRAD
Y=NFRAC
FRACI=FRAC-Y
DAMFRA=FRACI*DAMRAD
PRINT*, " FRACI= ",FRACI
PRINT*, " DAMFRA= ",DAMFRA

C EFFECTIVE RUNWAY WIDTH (ERUNWI)

ERUNWI=RUNWID+2.*DAMRAD
ERWID=RUNWID+DAMRAD

C NUMBER OF SECTIONS PER GROUP (NSPG)

$NSPG = (DAMRAD + TOWID + DAMFRA) / STEP + 1$

C NUMBER OF SECTIONS PER WIDTH (NSPW)

$NSPW = ERWID / STEP + 1$

C NUMBER OF GROUPS PER WIDTH (NGPW)

$NGPW = NSPW - NSPG + 1$

C WORKING WITH THE "FRACTION"

$FRA = (DAMRAD + TOWID + DAMFRA) / STEP$

$NFRA = (DAMRAD + TOWID + DAMFRA) / STEP$

$Y = NFRA$

$FRAI = FRA - Y$

RETURN

END

Subroutine ITER

SUBROUTINE ITER

C*****

C THIS SUBROUTINE (ITER) CALCULATES THE NUMBER OF
C ITERATIONS (NITERA) WHICH ARE REQUIRED BY THE SIMULATION
C FOR ONE TO BE 99% CONFIDENT THAT THE ACTUAL PROBABILITY OF
C MAKING THE CUT IS WITHIN .01 OF THE VALUE COMPUTED

C*****

COMMON/FIXED/RUNWID,DAMRAD,STADEV,TOWID,PROCUT
COMMON/ITERAT/NITERA

C*****

X=((2.576**2.)*(PROCUT)*(1.-PROCUT))/((.01)**2.)
NITERA=X+1.
RETURN
END

Subroutine BOUNDS

SUBROUTINE BOUNDS(MIN,MAX,NUMAIM)

C*****

C THIS SUBROUTINE (BOUNDS) COMPUTES THE MINIMUM LOWER
C BOUNDS ON THE WEAPON AND AN INITIAL UPPER BOUND. THESE
C BOUNDS ARE ON THE NUMBER OF WEAPONS REQUIRED TO MAKE
C A CUT WITH THE GIVEN DESTRUCTION CRITERIA (IN THIS
C CASE PROCUT)
C THE FOLLOWING VARIABLES ARE:
C 1. MIN=THE MINIMUM NUMBER OF WEAPONS REQUIRED (THIS
C NUMBER COMES FROM USING AN AIM POINT STRATEGY
C WHICH WILL GUARANTEE THAT NO SECTION OF LENGTH TOWID
C WILL REMAIN UNDAMAGED. THEN MIN IS THE NUMBER
C AIM POINTS FOR THE STRATEGY (ONE IS ASSUMING THAT
C THE CEP IS 0)
C 2. MAX=THE NUMBER OF WEAPONS REQUIRED IF THE MIN
C AIM POINT STRATEGY IS USED WITH THE ACTUAL CEP
C OF THE WEAPON UNDER INVESTIGATION
C 3. TMARGIN IS AN ARRAY FOR USE IN COMPARING THE
C MARGINAL IMPROVEMENT IN THE PROBABILITY OF
C MAKING THE CUT WITH ONE MORE WEAPON ADDED TO
C EACH AIM POINT. THIS IS DONE SEPARATELY SO THAT
C ON NO ONE ITERATION WILL THE TOTAL NUMBER
C OF WEAPONS INCREASE BY MORE THEN ONE
C 4. PROMIN=THE PROBABILITY OF MAKING THE CUT
C WITH THE MIN AIM POINT STRATEGY
C 5. AIMLOC=AN ARRAY OF THE AIM POINT LOCATIONS
C 6. EPSILN=A CONVERGENCE CRITERIA FOR SUBROUTINE UNION
C WHICH IS SET TO ZERO FOR THE USE IN THIS THESIS
C 7. NJMAIM=THE NUMBER OF AIM POINTS

C*****

DIMENSION AIMLOC(20),NUMBOM(20),TMARGIN(20)
DIMENSION MINBMB(20),MINAP(20)
COMMON/MARGIN/TMARGIN
COMMON/PROSPG(12,20,20)
COMMON/FIXED/RUNWID,DAMRAD,STADEV,TOWID,PROCUT
COMMON/GRP/NSPG,NSPW,NGPW,STEP,FRAI,ERUNWI,ERWID
COMMON/EQUAL/MINAP,MINNBR,MINBMB,NUMBER

C*****

```

J10=0
IF((NGPW.GT.12).AND.(PROCUT.LT..99)) J10=1
IF((NGPW.GT.12).AND.(PROCUT.LT..99)) PROCUT=PROCUT+.01
EPSILN=0.
N=1
  AIMLOC(1)=TOWID+DAMRAC
  N1=RUNWIC/(TOWID+DAMRAD)
  X2=RUNWID/(TOWID+DAMRAD)
  IF(X2.LT.1.) N=1
  IF(X2.LT.1.) GO TO 11
  IF((X2.LT.2.).AND.(X2.GE.1.)) N=1
  IF((X2.LT.2.).AND.(X2.GE.1.)) GO TO 11
  IF(X2.EQ.2.) GO TO 11
  IF(N1.EQ.2) N=2
  IF(N1.EQ.2) GO TO 11
  DO 10 I=2,N1
  X=I-1
  AIMLOC(I)=AIMLOC(1)+X*(TOWID+2.*DAMRAD)
  IF(AIMLOC(I).GT.RUNWID) AIMLOC(I)=
CAIMLOC(1)+X*(TOWID+2.*DAMRAD)+TOWID+DAMRAD
  IF(AIMLOC(I).GT.RUNWID) AIMLOC(I)=0.
  IF(AIMLOC(I).GT.RUNWID) GO TO 11
  N=N+1
10  CONTINUE
11  CONTINUE
  NUMAIM=N
  W=NUMAIM
  X=NUMAIM+1
  Y=RUNWID/X
  Z=W/2.
  N8=W/2.
  Z1=Z-N8
  IF(Z1.GT.0.) GO TO 50
  DO 12 I=1,N8
  X1=I
  AIMLOC(I)=X1*Y
12  CONTINUE
  N9=N8+1
  DO 60 I=N9,N
  X1=N-I+1
  AIMLOC(I)=RUNWID-X1*Y
60  CONTINUE
  GO TO 70
50  CONTINUE
  DO 51 I=1,N8
  X1=I
  AIMLOC(I)=X1*Y
51  CONTINUE

```

```

      N9=N8+2
      DO 52 I=N9,N
      X1=N-I+1
      AIMLOC(I)=RUNWID-X1*Y
52    CONTINUE
      CENTER=RUNWID/2.
      AIMLOC(N8+1)=CENTER
70    CONTINUE
      DO 13 I=1,NUMAIM
      NUMBOM(I)=1
13    CONTINUE
      IF(NGPW.GT.12) GO TO 20
      CALL PROCAL(NUMAIM,AIMLOC,NUMBOM)
      CALL UNION(NUMBOM,NUMAIM,EPSILN,TOTAL)
      GO TO 21
20    CONTINUE
      CALL SIM(NUMAIM,AIMLOC,NUMBOM,TOTAL)
21    CONTINUE
      MIN=NUMAIM
      PROMIN=1-TOTAL
      IF(PROMIN.GE.PROCUT) MAX=MIN
      IF(PROMIN.GE.PROCUT) GO TO 100
      DO 14 I=1,50
      DO 15 J=1,NUMAIM
      NUMBOM(J)=NUMBOM(J)+1
      IF(NGPW.GT.12) GO TO 30
      CALL PROCAL(NUMAIM,AIMLOC,NUMBOM)
      CALL UNION(NUMBOM,NUMAIM,EPSILN,TOTAL)
      GO TO 31
30    CONTINUE
      CALL SIM(NUMAIM,AIMLOC,NUMBOM,TOTAL)
31    CONTINUE
      NUMBOM(J)=NUMBOM(J)-1
      TMARGIN(J)=1.-TOTAL
15    CONTINUE
      N=N+1
      X=X+1.
      DO 16 K=1,NUMAIM
      IF(X.LT.TMARGIN(K)) N=N+1
      IF(X.LT.TMARGIN(K)) X=TMARGIN(K)
16    CONTINUE
      NUMBOM(N)=NUMBOM(N)+1
      IF(TMARGIN(N).GT.PROCUT) GO TO 100
      IF(I.EQ.50) PRINT*," NOT ENOUGH TOTAL WEAPONS"
      IF(I.EQ.50) GO TO 200
14    CONTINUE
100   CONTINUE

```

```

      N9=N8+2
      DO 52 I=N9, N
      X1=N-I+1
      AIMLOC(I)=RUNWID-X1*Y
52    CONTINUE
      CENTER=RUNWID/2.
      AIMLOC(N8+1)=CENTER
70    CONTINUE
      DO 13 I=1, NUMAIM
      NUMBOM(I)=1
13    CONTINUE
      IF(N3PW.GT.12) GO TO 20
      CALL PROCAL(NUMAIM, AIMLOC, NUMBOM)
      CALL UNION(NUMBOM, NUMAIM, EPSILN, TOTAL)
      GO TO 21
20    CONTINUE
      CALL SIM(NUMAIM, AIMLOC, NUMBOM, TOTAL)
21    CONTINUE
      MIN=NUMAIM
      PROMIN=1-TOTAL
      IF(PROMIN.GE.PROCUT) MAX=MIN
      IF(PROMIN.GE.PROCUT) GO TO 100
      DO 14 I=1, 50
      DO 15 J=1, NUMAIM
      NUMBOM(J)=NUMBOM(J)+1
      IF(N3PW.GT.12) GO TO 30
      CALL PROCAL(NUMAIM, AIMLOC, NUMBOM)
      CALL UNION(NUMBOM, NUMAIM, EPSILN, TOTAL)
      GO TO 31
30    CONTINUE
      CALL SIM(NUMAIM, AIMLOC, NUMBOM, TOTAL)
31    CONTINUE
      NUMBOM(J)=NUMBOM(J)-1
      TMARGIN(J)=1.-TOTAL
15    CONTINUE
      N=0
      X=0.
      DO 16 K=1, NUMAIM
      IF(X.LT.TMARGIN(K)) N=N+1
      IF(X.LT.TMARGIN(K)) X=TMARGIN(K)
16    CONTINUE
      NUMBOM(N)=NUMBOM(N)+1
      IF(TMARGIN(N).GT.PROCUT) GO TO 100
      IF(I.EQ.50) PRINT*, " NOT ENOUGH TOTAL WEAPONS"
      IF(I.EQ.50) GO TO 200
14    CONTINUE
100   CONTINUE

```

```
MINN3R=0
DO 103 I=1,NUMAIM
MINBMB(I)=NUMBOM(I)
MINAP(I)=ATMLOC(I)
MINN3R=MINNBR+MINBMB(I)
103 CONTINUE
N=0
DO 17 I=1,NUMAIM
N=N+NUMBOM(I)
17 CONTINUE
MAX=N
200 CONTINUE
IF(J10.EQ.1) PROCUT=PROCUT-.01
RETURN
END
```

Subroutine COUNT

SUBROUTINE COUNT(ICOUNT,MINNUM)

C*****

C THIS SUBROUTINE (COUNT) ALLOWS ONE TO VARY THE NUMBER
C OF WEAPONS TARGETED FOR THE AIM POINTS. IT IS USED
C IN SUBROUTINE STRAT TO DETERMINE THE BEST AIM POINT
C STRATEGY FOR A GIVEN NUMBER OF AIM POINTS WITH
C ICOUNT WEAPONS TARGETED FOR EACH AIM POINT, I.E.
C EVERY AIM POINT IN STRAT HAS THE SAME NUMBER OF
C WEAPONS TARGETED FOR IT.

C THE FOLLOWING VARIABLES ARE:

- C 1. MINNBR=THE CURRENT UPPER BOUND ON THE NUMBER
C OF WEAPONS REQUIRED TO MAKE THE CUT WITH THE
C DESIRED PROCUT
- C 2. MINAP=AN ARRAY OF THE LOCATIONS OF THESE AIM POINTS
C WHICH ACHEIVE THE CURRENT UPPER BOUND ON THE
C NUMBER OF WEAPONS REQUIRED
- C 3. MINBMB=AN ARRAY OF THE NUMBER OF WEAPONS TARGETED
C FOR EACH AIM POINT IN MINAP
- C 4. PROB=THE CURRENT PROBABILITY OF THE "BEST" AIM POINT STRATEGY
- C 5. THRGIN=AN ARRAY COMPUTED IN SUBROUTINE BEST THAT HAS THE
C CURRENT PROBABILITIES FOR EACH AIM POINT GROUPING

C*****

DIMENSION MINAP(20),APMIN(20),MINBMB(20),MINBOM(20)
DIMENSION THRGIN(20),VALUE(20),NUMBOM(20)
DIMENSION BESTAP(20,20)
DIMENSION IND(20)
DIMENSION BVALUE(20)
COMMON/MARGIN/THRGIN
COMMON/OPT/BESTAP,IND,BVALUE
COMMON/EQUAL/MINAP,MINNBR,MINBMB,NUMBER

C*****

```

10  CONTINUE
    ICOUNT=ICOUNT+1
    X=ICOUNT
    Y=MINNUM
    MAX=MINNUM
    ISTOP=Y/X
    IF(ISTOP.LE.1) GO TO 100
    CALL STRAT(ICOUNT,MAX,VALUE,NSET)
    CALL BEST(ICOUNT,MINNBR,MAX,NUMBOM,NSET,APMIN,MINBOM,MINNUM,NGP)

```

C J45 IS A COUNTER WHICH SETS THE INITIAL VALUE OF PROB,
 C DEPENDING ON WHETHER ICOUNT STARTS AT 1 OR 2 IN THIS
 C SUBROUTINE

```

    IF(MINNBR.EQ.MINNUM) GO TO 60
    J45=0
    IF(ICOUNT.EQ.1) J45=1
    IF(J45.EQ.1) PROB=0.
    IF((ICOUNT.EQ.2).AND.(J45.EQ.0)) PROB=0.
    IF(PROB.LT.TMRGIN(NGP)) GO TO 50
    GO TO 10
50  CONTINUE
    PROB=TMRGIN(NGP)
60  CONTINUE
    NUMBER=NGP
    MINNJM=0
    DO 51 I=1,NGP
    MINNUM=MINNUM+MINBOM(I)
    MINAP(I)=APMIN(I)
    MINBMB(I)=MINBOM(I)
51  CONTINUE
    IF(MINNUM.EQ.MINNBR) GO TO 100
    GO TO 10
100 CONTINUE
    RETURN
    END

```

Subroutine STRAT

SUBROUTINE STRAT(ICOUNT,MAX,VALUE,NSET)

C*****

C THIS SUBROUTINE (STRAT) DETERMINES THE BEST AIM POINT
C STRATEGY FOR A GIVEN NUMBER OF AIM POINTS AND THE FACT
C THAT THESE AIM POINTS HAVE THE SAME NUMBER OF WEAPONS
C TARGETED FOR EACH OF THEM. THESE STRATEGIES ARE
C SAVED IN THE BESTAP MATRIX AND THEN ARE PASSED TO
C SUBROUTINE BEST FOR DETERMINING WHICH OF THE THESE
C AIM POINT GROUPS IS ACTUALLY THE BETTER AIM POINT
C STRATEGY

C THIS SUBROUTINE WORKS ON THE IDEA THAT IF ALL AIM
C POINTS EXCEPT THE MIDDLE TWO ARE HELD FIXED THE
C RESULTING PROBABILITY DISTRIBUTION SHOULD BE
C CONVEX. THUS WHEN THE PROBABILITY OF NOT MAKING
C THE CUT BEGINS TO INCREASE ONE HAS PASSED THE BEST
C LOCATION FOR THIS CONFIGURATION OF AIM POINTS
C WITH AN ODD NUMBER OF AIM POINTS, THE CENTER AIM
C POINT IS ALSO HELD FIXED

C THE FOLLOWING VARIABLES ARE:

- C 1. EPSILN=A CONVERGENCE CRITERIA FOR SUBROUTINE UNION
C WHICH IS SET TO ZERO FOR THE USE IN THIS THESIS
- C 2. NSET=THE CURRENT NUMBER OF AIM POINT GROUPINGS
- C 3. CENAIM=THE CENTER OF THE ACTUAL RUNWAY (CORRESPONDS TO THE
C MIDDLE AIM POINT FOR AN ODD AIM POINT GROUP)
- C 5. AIMGP=AN ARRAY INDEXED BY AIM POINT GROUPING AND THE AIM
C POINT NUMBER WITHIN THAT SPECIFIC GROUPING
- C 6. NSTEP=THE NUMBER OF STEPS (BY FIVE FEET) FROM THE INITIALIZED
C POSITIONS OF THE INNER MOST PAIR OF SYMMETRIC AIM POINTS
- C 7. VALUE=AN ARRAY OF THE CURRENT PROBABILITIES OF NOT MAKING
C THE CUT. WHENEVER THIS PROBABILITY DECREASES FOR A SPECIFIC
C AIM POINT GROUP, THE NEW PROBABILITY IS PLACED IN VALUE AND
C THEN THIS PROBABILITY IS COMPARED TO THE CURRENT PROBABILITY
C STORED IN BVALUE. IF VALUE IS LESS THAN BVALUE, THEN THIS AIM
C POINT STRATEGY IS STORED IN BESTAP BY AIM POINT GROUP NUMBER
C AND THE LOCATION OF EACH MEMBER OF THIS GROUP

C*****

```
DIMENSION AIMLOC(20),NUMBOM(20),AIMGP(20,20),INDX(20),
CBESTAP(20,20),VALUE(20),IND(20),BVALUE(20)
COMMON PROSPG(12,20,20)
COMMON/AIMGRP/AIMGP
COMMON/FIXED/RUNWID,DAPRAD,STADEV,TOWID,PROCUT
COMMON/GRP/NSPG,NSPW,NGPW,STEP,FRAI,ERUNWI,ERWID
COMMON/OPT/BESTAP,IND,EVALUE
```

C*****

C DETERMINE THE UPPER BOUND ON THE NUMBER OF AIM POINTS

```
EPSILN=0.
NSET=MAX/ICOUNT
```

C DETERMINE THE "BEST" AIM POINT STRATEGY FOR EACH AIM

C POINT SET

```
      K=0
10    CONTINUE
      K=K+1
      IF(K.GT.NSET) GO TO 1000
      CENAIM=RUNWID/2.
      IF(K.GT.1) GO TO 20
      AIMGP(1,1)=CENAIM
      GO TO 10
20    CONTINUE
```

C DETERMINE WHETHER ONE IS DEALING WITH AN ODD OR AN EVEN

C AIM POINT SET

```
X=K
N=X/2.
Y=X/2.
Z=Y-N
IF(Z.EQ.0) GO TO 30
```

C INITIALIZE AN ODD AIM POINT SET

```

      AIMGP(K,N+1)=CENAIM
      N1=N+1
      DO 21 I=1,N
      X=I
      AIMGP(K,I)=X*DAMRAD
21    CONTINUE
      M1=N+2
      DO 22 I=M1,K
      X=K-I+1.
      AIMGP(K,I)=RUNWID-X*DAMRAD
22    CONTINUE
      GO TO 40
30    CONTINUE

```

C INITIALIZE AN EVEN AIM POINT SET

```

      N1=N
      DO 31 I=1,N
      X=I
      AIMGP(K,I)=X*DAMRAD
31    CONTINUE
      M1=N+1
      DO 32 I=M1,K
      X=K-I+1.
      AIMGP(K,I)=RUNWID-X*DAMRAD
32    CONTINUE
40    CONTINUE

```

C THE ITERATIVE SCHEME TO DETERMINE THE "BEST" AIM POINT
C STRATEGY PER AIM POINT SET

```

      DO 41 I=1,K
      AIMLOC(I)=AIMGP(K,I)
      NUMBOM(I)=ICOUNT
41    CONTINUE
      NUMAIM=K
      IF(NGPW.GT.12) GO TO 50
      CALL PROCAL(NUMAIM,AIMLOC,NUMBOM)
      CALL UNION(NUMBOM,NUMAIM,EPSILN,TOTAL)
      GO TO 51
50    CONTINUE
      CALL SIM(NUMAIM,AIMLOC,NUMBOM,TOTAL)
51    CONTINUE
      VALUE(K)=TOTAL
      DO 42 I=1,K
      BESTAP(K,I)=AIMGP(K,I)
42    CONTINUE

```

```

J36=K-1
DO 60 I=1,J36
DIS=AIMGP(K,I+1)-AIMGP(K,I)
IF(DIS.LE.DAMRAD) GO TO 70
60 CONTINUE
GO TO 80
70 CONTINUE
DO 71 I=1,K
BESTAP(K,I)=0.
71 CONTINUE
GO TO 10
80 CONTINUE
IF(N1.EQ.N) GO TO 200

```

C WORKING WITH THE ODD AIM POINT SETS

L=N1-1

C L IS AN INDEX FOR DETERMINING WHICH PAIR OF SYMMETRIC
C AIM POINTS IS BEING MOVED. THE SMALLER L IS, THE CLOSER
C ONE IS TO THE MIDDLE PAIR, WITH L=1 BEING THE MIDDLE PAIR.

```

DO 101 I=1,L
IND(I)=I
101 CONTINUE
BVALUE(K)=1.
L1=1
105 CONTINUE
IF(L1.GT.L) GO TO 10
110 CONTINUE
J21=0

```

C J21 IS A COUNTER ON THE NUMBER OF TIMES THE AIM POINT
C STRATEGY EXCEEDS THE DIFFERENCE CRITERIA

```

Y=L1
VALUE(K)=1.
IF(K.EQ.3) GO TO 150
AIMGP(K,N1-IND(I))=AIMGP(K,N1-IND(I))+5.
AIMGP(K,N1+IND(I))=AIMGP(K,N1+IND(I))-5.

```

C MOVES EACH OF THE AIM POINTS DESIGNATED BY L1 (I) 5 FEET
C INWARD

```

J1=N1-IND(I)+1
J2=N1+1
J4=N1-1
X=0.

```

C INITIALIZES ALL AIM POINTS INBOARD OF THE TWO JUST MOVED
C (ALL PAIRS WITH L1<I)

```

DO 112 M2=J1,J4
X=X+1.
AIMGP(K,M2)=AIMGP(K,N1-IND(I))+X*DAMRAD
112 CONTINUE
IF(AIMGP(K,J4).GE.CENAIM) L1=L1+1
IF(AIMGP(K,J4).GE.CENAIM) GO TO 105
IF((CENAIM-AIMGP(K,J4)).LT.DAMRAD) L1=L1+1
IF((CENAIM-AIMGP(K,J4)).LT.DAMRAD) GO TO 105
J3=N1+IND(I)-1
DO 113 M2=J2,J3
X=K-M2
AIMGP(K,M2)=AIMGP(K,IND(I)+N1)-X*DAMRAD
113 CONTINUE
GO TO 150
120 CONTINUE
I=I+1
IF(I.GT.L) GO TO 10
GO TO 110
150 CONTINUE
NSTEP=(CENAIM-AIMGP(K,N1-1))/5.
I=1
151 CONTINUE
I=I+1
IF(I.EQ.2) GO TO 155
IF(I.GT.NSTEP) GO TO 154
AIMGP(K,N1-1)=AIMGP(K,N1-1)+5.

```

C CHECK TO MAKE SURE THAT THE CENTER PAIR OF AIM POINTS
C (1) HAVE NOT CROSSED THE CENTER POINT OF THE RUNWAY
C (2) ARE NOT LESS THAN ONE DAMAGE RADIUS FROM THE CENTER
C OF THE RUNWAY

```

IF(AIMGP(K,N1-1).GT.CENAIM) L1=L1+1
IF(AIMGP(K,N1-1).GT.CENAIM) GO TO 105
IF((CENAIM-AIMGP(K,N1-1)).LT.DAMRAD) L1=L1+1
IF((CENAIM-AIMGP(K,N1-1)).LT.DAMRAD) GO TO 105
AIMGP(K,N1+1)=AIMGP(K,N1+1)-5.
155 CONTINUE
DO 152 I1=1,K
AIMLOC(I1)=AIMGP(K,I1)
152 CONTINUE

```

C ALLOWS THE FIRST OCCURRENCE OF DIF EXCEEDING IT'S CRITERIA
 C TO NOT STOP THE ITERATIVE PROCESS. THE SECOND TIME DIF EXCEEDS
 C ITS CRITERIA, THE CURRENT BASE PAIR OF AIM POINTS (L1) IS MOVE
 C 5 FEET INBOARD AND THE ITERATIVE PROCESS STARTS AGAIN

```

    IF(I.GT.3) CALL APROX(AIMLOC,NUMAIM,NUMBOM,DIF)
    IF(I.LE.3) DIF=.01
    IF(DIF.LT.0.) PRINT*,"STARTED TOO CLOSE TO THE EDGE"
    IF(DIF.GT..05) J21=J21+1
    IF((J21.GT.1).AND.(K.EQ.3)) L1=L1+1
    IF((J21.GT.1).AND.(K.EQ.3)) GO TO 105
    IF(J21.GT.1) GO TO 110
    IF(NGPW.GT.12) GO TO 160
    CALL PROCAL(NUMAIM,AIMLOC,NUMBOM)
    CALL UNION(NUMBOM,NUMAIM,EPSILN,TOTAL)
    GO TO 161
160  CONTINUE
    CALL SIN(NUMAIM,AIMLOC,NUMBOM,TOTAL)
161  CONTINUE
    IF((NGPW.GT.12).AND.(I.EQ.2)) GO TO 163
  
```

C CHECK TO SEE IF TOTAL, THE PROBABILITY OF NOT MAKING
 C A CUT, IS STILL DECREASING. IF IT IS, RESET VALUE(K) TO
 C TOTAL. IF IT IS NOT, THEN MUST MOVE THE CURRENT PAIR (L1)
 C 5 MORE FEET INBOARD AND ITERATE AGAIN.

```

    IF((VALUE(K).LT.TOTAL).AND.(K.EQ.3)) L1=L1+1
    IF((VALUE(K).LT.TOTAL).AND.(K.EQ.3)) GO TO 105
    IF((VALUE(K).LT.TOTAL).AND.(I.EQ.3)) L1=L1+1
    IF((VALUE(K).LT.TOTAL).AND.(I.EQ.3)) GO TO 105
    IF(VALUE(K).LT.TOTAL) GO TO 110
163  CONTINUE
    VALUE(K)=TOTAL
    IF(VALUE(K).GT.BVALUE(K)) GO TO 151
    BVALUE(K)=VALUE(K)
    DO 153 J10=1,K
    BESTAP(K,J10)=AIMGP(K,J10)
153  CONTINUE
    GO TO 151
154  CONTINUE
    PRINT*," IS CONTINUALLY DECREASING ON THIS INTERVAL. K= ",K
    L1=L1+1
    GO TO 105
  
```

200 CONTINUE

C WORKING WITH THE EVEN AIM POINT SETS

L=N1

C L IS AN INDEX FOR DETERMINING WHICH PAIR OF SYMMETRIC
C AIM POINTS IS BEING MOVED. THE SMALLER L IS, THE CLOSER
C ONE IS TO THE MIDDLE PAIR, WITH L=1 BEING THE MIDDLE PAIR.

```
DO 201 I=1,L
  IND(I)=I
201 CONTINUE
  BVALJE(K)=1.
  L1=1
205 CONTINUE
  IF(L1.GT.L) GO TO 10
210 CONTINUE
  I=L1
  VALUE(K)=1.
```

C J22 IS A COUNTER ON THE NUMBER OF TIMES THE AIM POINT
C STRATEGY EXCEEDS THE DIFFERENCE CRITERIA

```
J22=0
IF(K.EQ.2) GO TO 250
IF(L1.EQ.1) GO TO 250
```

C MOVES EACH OF THE AIM POINTS DESIGNATED BY L1 (I) 5 FEET
C INWARD

```
AIMGP(K,N1+1-IND(I))=AIMGP(K,N1+1-IND(I))+5.
AIMGP(K,N1+IND(I))=AIMGP(K,N1+IND(I))-5.
J1=N1+2-IND(I)
IF(L1.EQ.1) J1=N1
X=0.
```

C INITIALIZES ALL AIM POINTS INBOARD OF THE TWO JUST MOVED
C (ALL PAIRS WITH L1<I)

```
DO 212 M2=J1,N1
  X=X+1.
  AIMGP(K,M2)=AIMGP(K,N1+1-IND(I))+X*DAHRA
212 CONTINUE
  IF(AIMGP(K,N1).GE.CENAIM) L1=L1+1
  IF(AIMGP(K,N1).GE.CENAIM) GO TO 205
  J2=N1+1
  J3=N1+IND(I)-1
  IF(L1.EQ.1) J3=N1+1
  DO 213 M2=J2,J3
  X=X-M2
  AIMGP(K,M2)=AIMGP(K,IND(I)+N1)-X*DAHRA
213 CONTINUE
```

C CHECK TO MAKE SURE THAT THE CENTER PAIR OF AIM POINTS
 C (1) HAVE NOT CROSSED THE CENTER POINT OF THE RUNWAY
 C (2) ARE NOT LESS THAN ONE DAMAGE RADIUS APART

```

      IF(AIMG(K,N1).GE.AIMG(K,J2)) L1=L1+1
      IF(AIMG(K,N1).GE.AIMG(K,J2)) GO TO 205
      DIS=AIMG(K,J2)-AIMG(K,N1)
      IF(DIS.LE.DAMRAD) L1=L1+1
      IF(DIS.LE.DAMRAD) GO TO 205
      GO TO 250
220  CONTINUE
      I=I+1
      IF(I.GT.L) GO TO 10
      GO TO 210
250  CONTINUE
      NSTEP=(CENAIM-AIMG(K,N1))/5.
      I=1
251  CONTINUE
      I=I+1
      IF(I.EQ.2) GO TO 255
      IF(I.GT.NSTEP) GO TO 254
      AIMG(K,N1)=AIMG(K,N1)+5.
      AIMG(K,N1+1)=AIMG(K,N1+1)-5.
255  CONTINUE
      J=N1+1
      IF(AIMG(K,N1).GE.AIMG(K,N1+1)) L1=L1+1
      IF(AIMG(K,N1).GE.AIMG(K,N1+1)) GO TO 205
      DIS=AIMG(K,N1+1)-AIMG(K,N1)
      IF(DIS.LE.DAMRAD) L1=L1+1
      IF(DIS.LE.DAMRAD) GO TO 205
      DO 252 J=1,K
      AIMLOC(J)=AIMG(K,J)
252  CONTINUE

```

C ALLOWS THE FIRST OCCURRENCE OF DIF EXCEEDING IT'S CRITERIA
 C TO NOT STOP THE ITERATIVE PROCESS. THE SECOND TIME DIF EXCEEDS
 C ITS CRITERIA, THE CURRENT BASE PAIR OF AIM POINTS (L1) IS MOVE
 C 5 FEET INBOARD AND THE ITERATIVE PROCESS STARTS AGAIN

```

      IF(I.GT.3) CALL APROX(AIMLOC,NUMAIM,NUMBOM,DIF)
      IF(I.LE.3) DIF=.01
      IF(DIF.LT.0.) PRINT*, "STARTED TOO CLOSE TO THE EDGE"
      IF(DIF.GT..05) J22=J22+1
      IF((J22.GT.1).AND.(K.EQ.2)) L1=L1+1
      IF((J22.GT.1).AND.(K.EQ.2)) GO TO 205
      IF(J22.GT.1) GO TO 210

```

```

      IF(NGPW.GT.12) GO TO 260
      CALL PROCAL(NUMAIM,AIMLOC,NUMBOM)
      CALL UNION(NUMBOM,NUMAIM,EPSILN,TOTAL)
      GO TO 261
260  CONTINUE
      CALL SIM(NUMAIM,AIMLOC,NUMBOM,TOTAL)
261  CONTINUE
      IF((NGPW.GT.12).AND.(I.EQ.2)) GO TO 263

C CHECK TO SEE IF TOTAL, THE PROBABILITY OF NOT MAKING A CUT,
C IS STILL DECREASING. IF IT IS, RESET VALUE(K) TO TOTAL.
C IF IT IS NOT, THEN MUST MOVE THE CURRENT PAIR (L1) 5
C MORE FEET INBOARD AND ITERATE AGAIN.

      IF((VALUE(K).LT.TOTAL).AND.(K.EQ.2)) L1=L1+1
      IF((VALUE(K).LT.TOTAL).AND.(K.EQ.2)) GO TO 205
      IF((VALUE(K).LT.TOTAL).AND.(I.EQ.3)) L1=L1+1
      IF((VALUE(K).LT.TOTAL).AND.(I.EQ.3)) GO TO 205
      IF(VALUE(K).LT.TOTAL) GO TO 210
263  CONTINUE
      VALUE(K)=TOTAL
      IF(VALUE(K).GT.BVALUE(K)) GO TO 251
      BVALUE(K)=VALUE(K)
      DO 253 J10=1,K
      BESTAP(K,J10)=AIMGP(K,J10)
253  CONTINUE
      GO TO 251
254  CONTINUE
      PRINT*, " IS CONTINUALLY DECREASING ON THIS INTERVAL. K= ",K
      L1=L1+1
      GO TO 205
1000 CONTINUE
      RETURN
      END

```

Subroutine APROX

SUBROUTINE APROX(AIMLOC,NUMAIM,NUMBOM,DIF)

```

C*****
C THIS SUBROUTINE (APROX) COMPARES THE PROBABILITY OF DAMAGING
C THE MIDDLE GROUP WITH THE PROBABILITY OF DAMAGING THE
C FIRST GROUP. THIS ALLOWS FOR A QUICK CHECK ON THE AIM
C POINT STRATEGY UNDER CONSIDERATION IN SUBROUTINE STRAT.
C THE FOLLOWING VARIABLES ARE:
C   1. DAMLOC=AN ARRAY OF THE AIM POINT LOCATIONS RELATIVE TO THE
C      EFFECTIVE RUNWAY WIDTH
C   2. CENTER=CENTER OF THE ACTUAL RUNWAY WIDTH
C   3. GRPLEN=LENGTH OF A GROUP
C   4. HAFLEN=HALF OF GRPLEN
C   5. MIDGRP=THE STARTING POINT OF THE "MIDDLE" GROUP
C   6. EQUIPRO=AN ARRAY OF THE SECTION BOUNDARIES IN TERMS OF
C      STANDARD DEVIATION FROM ITS AIM POINT. THE ARRAY IS INDEXED
C      BY GROUP NUMBER (WHERE 1 IS THE FIRST GROUP AND 2 IS THE MIDDLE
C      GROUP), AIM POINT NUMBER, AND SECTION NUMBER WITHIN EACH GROUP
C   7. MDNOR=AN IMSL SUBROUTINE THAT RETURNS P(X<A) FOR A N(0,1)
C      DISTRIBUTION
C   8. PROEQU=AN ARRAY OF THE PROBABILITIES THAT A WEAPON IMPACTS
C      WITHIN A SECTION. THE ARRAY IS INDEXED THE SAME AS EQUIPRO.
C*****
-----
      DIMENSION EQUIPRO(2,20,25),S1(25),PROD(2),PROEQU(2,20,25)
      DIMENSION DAMLOC(20),NUMBOM(20),AIMLOC(20)
      COMMON/FIXED/RUNWID,DARAD,STADEV,TOWID,PROCUT
      COMMON/GRP/NSPG,NSPW,NGPW,STEP,FRAI,ERUNWI,ERWID
      COMMON/APPROX/EQUIPRO,S1,PROD,PROEQU
C*****

```

AD-A094 807

AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCH00--ETC F/6 15/7
A GENERALIZED COMPUTER MODEL FOR THE TARGETING OF CONVENTIONAL --ETC(U)
DEC 80 J C PEMBERTON
AFIT/60R/05/80D-6

UNCLASSIFIED

NL

2 OF 2

AD-A094 807

END

DATE

FILED

3-81

DTIC

```

DO 1 I=1,NUMAIM
DAHLOC(I)=AIMLOC(I)+DAMRAD
1 CONTINUE
CENTER=RUNWID/2.
GRPLEN=FRAI*STEP+TCHID+DAMRAD
HAFLN=GRPLEN/2.
MIDGRP=CENTER-HAFLN
NSPG=1=NSPG+1
DO 2 I=1,2
DO 2 K=1,NUMAIM
DO 2 J=1,NSPG1
X=0.
IF(I.EQ.2) X=MIDGRP
Y=J-1
IF(I.EQ.NSPG) Y=Y+FRAI
EQUPRO(I,K,J)=((Y*STEP)+X-DAHLOC(K))/STADEV
2 CONTINUE
DO 3 I=1,2
DO 3 K=1,NUMAIM
DO 3 J=1,NSPG
CALL MDNOR(EQUPRO(I,K,J),BOND1)
L=J+1
CALL MDNOR(EQUPRO(I,K,L),BOND1)
PROEQU(I,K,J)=BOND1-BOND1
3 CONTINUE
DO 6 I=1,2
PROD(I)=1.
DO 4 J=1,NUMAIM
S1(J)=0.
DO 5 K=1,NSPG
S1(J)=S1(J)+PROEQU(I,J,K)
5 CONTINUE
S1(J)=S1(J)**NUMBOM(J)
PROD(I)=PROD(I)*S1(J)
4 CONTINUE
6 CONTINUE
DIF=PROD(2)-PROD(1)
RETURN
END

```

Subroutine BEST

SUBROUTINE BEST(ICOUNT,MIN,MAX,NUMBOM,NSET,APMIN,MINBOM,MINNUM,
CNGP)

C*****

C THIS SUBROUTINE (BEST) TAKES THE MOST "EFFICIENT" AIM POINTS
C STRATEGY PER AIM POINT GROUP (BESTAP) AND THEN, THROUGH
C MARGINAL ANALYSIS, DETERMINES WHICH OF THESE AIM POINT
C STRATEGIES REQUIRES THE MINIMUM NUMBER OF WEAPONS.
C THE FOLLOWING VARIABLES ARE:
C 1. APMIN=AN ARRAY OF THE LOCATIONS OF THE CURRENT "BEST" AIM
C POINT STRATEGIES
C 2. MINBOM=AN ARRAY OF THE NUMBER OF WEAPONS TARGETED FOR
C EACH SPECIFIC AIM POINT IS APMIN
C 3. EPSILN=A CONVERGENCE CRITERIA FOR SUBROUTINE UNION
C WHICH IS SET TO ZERO FOR THE USE IN THIS THESIS
C 4. NMAX=THE LATEST UPPER BOUND ON THE TOTAL NUMBER OF WEAPONS
C 5. TMARGIN=AN ARRAY USED TO COMPARED THE MARGINAL INCREASE
C IN THE PROBABILITIES OF MAKING A CUT GIVEN THAT ONE WEAPON IS
C ADDED TO AN AIM POINT
C 6. MINNUM=THE TOTAL NUMBER OF WEAPONS IN MINBOM
C 7. NSP=NB=COUNTERS TO INDICATE THE "BEST" AIM POINT GROUP.

C*****

DIMENSION BESTAP(20,20),NUMBOM(20),APMIN(20),MINBOM(20),
CTMARGIN(20),AIMLOC(20)
DIMENSION IND(20)
DIMENSION BVALUE(20)
COMMON PROSPG(12,20,20)
COMMON/MARGIN/TMARGIN
COMMON/FIXED/RUNWID,DAMRAD,STADEV,TOWID,PROGUT
COMMON/GRP/NSPG,NSPW,NGPW,STEP,FRAI,ERUNWI,ERWID
COMMON/OPT/BESTAP,IND,BVALUE

C*****

```

DO 1 I=1,20
  THRGIN(I)=0.
  APMIN(I)=0.
  MIN30M(I)=0
1  CONTINUE
  EPSLN=0.
  J21=0

```

C SETS THE CENTER OF THE RUNWAY AS THE AIM POINT IN AIM POINT GROUP 1

```

BESTAP(1,1)=RUNWID/2.
N6=0
L=0

```

C L IS A COUNTER ON THE NUMBER OF THE AIM POINT GROUP
C AFTER AIM POINT GROUP 1 IS CHECKED ONCE IT NEED NOT BE RECHECKED

```

IF(ICOUNT.EQ.1) THRGIN(1)=0.
IF(ICOUNT.EQ.1) GO TO 10
IF(ICOUNT.EQ.2) L=1
  NUMAIN=L
  IF(ICOUNT.EQ.2) GO TO 20
  IF(ICOUNT.GT.2) L=1
10  CONTINUE
  L=L+1
  NUMAIN=L
  IF(L.GT.NSET) GO TO 110
  J8=1

```

C SETS THE "BEST" AIM POINTS FOR EACH AIM POINT GROUP,
C AS DETERMINED IN STRAT , INTO AIMLOC FOR USE IN EITHER PROCAL OR SIM
C SETS THE INITIAL ALLOCATION OF WEAPONS TO EACH AIM POINT AT ICOUNT

```

IF(BESTAP(L,1).EQ.0.) GO TO 10
DO 11 I=J8,L
  AIMLOC(I)=BESTAP(L,I)
  NUMBOM(I)=ICOUNT
11  CONTINUE

```

C SETS THE MAX NUMBER OF TOTAL WEAPONS ALLOCATED TO MAX-ICOUNT*L
C STOPS WHEN NO MORE WEAPONS CAN BE ALLOCATED WITHOUT EXCEEDING MAX

```

I=MAX-(L*ICOUNT)
IF(I.EQ.0) GO TO 30

```

C DETERMINES THROUGH MARGINAL ANALYSIS WHICH AIM POINT LOCATION WITHIN
 C AN AIM POINT GROUP WILL DECREASE TOTAL THE QUICKEST, GIVEN THAT ONLY
 C ONE MORE WEAPON IS ADDED TO ANY OF THE AIM POINT LOCATIONS

```

    DO 12 J=1,I
    DO 13 J1=J8,L
    NUMBOM(J1)=NUMBOM(J1)+1
    IF(NGPW.GT.12) GO TO 50
    CALL PROCAL(NUMAIM,AIMLOC,NUMBOM)
    CALL UNION(NUMBOM,NUMAIM,EPSILN,TOTAL)
    GO TO 51
50  CONTINUE
    CALL SIM(NUMAIM,AIMLOC,NUMBOM,TOTAL)
51  CONTINUE
    NUMBOM(J1)=NUMBOM(J1)-1
    TMARGIN(J1)=1.-TOTAL
    IF(NGPW.GT.12) TMARGIN(J1)=1.01-TOTAL
13  CONTINUE
    N=0
    X=0.
    DO 14 K=J8,L
    IF(X.LT.TMARGIN(K)) N=N+1
    IF(X.LT.TMARGIN(K)) X=TMARGIN(K)
14  CONTINUE
    NUMBOM(N)=NUMBOM(N)+1
    IF(TMARGIN(N).GE.PROCUT) GO TO 15
12  CONTINUE
    GO TO 10
30  CONTINUE
    IF(NGPW.GT.12) GO TO 31
    CALL PROCAL(NUMAIM,AIMLOC,NUMBOM)
    CALL UNION(NUMBOM,NUMAIM,EPSILN,TOTAL)
    GO TO 32
31  CONTINUE
    CALL SIM(NUMAIM,AIMLOC,NUMBOM,TOTAL)
32  CONTINUE
    TMARGIN(L)=1.-TOTAL
    IF(NGPW.GT.12) TMARGIN(L)=1.01-TOTAL
    IF(TMARGIN(L).GE.PROCUT) GO TO 15
    GO TO 10
15  CONTINUE

```

C COUNTS UP THE NUMBER OF TOTAL WEAPONS USED

```

    NMAX=0
    DO 16 K=J8,L
    NMAX=NMAX+NUMBOM(K)
16  CONTINUE
    IF((TMARGIN(L).LT.TMARGIN(1)).AND. (NMAX.EQ.MAX).AND.J21.EQ.1)
    C GO TO 10

```

C IF THE TOTAL NUMBER OF WEAPONS USED IS LESS THAN THE CURRENT UPPER
 C BOUND THIS AIM POINT STRATEGY IS STORED IN APHIN AND MINBOM
 C AND MAX IS NOW SET TO THE NEW UPPER BOUND, NMAX

```

    IF(NMAX.LT.MAX) N6=L
    IF(NMAX.LT.MAX) MAX=NMAX
    IF(MAX.EQ.MIN) GO TO 100
    IF(NMAX.EQ.MAX) N6=L
    IF(NMAX.EQ.MAX) GO TO 100
    GO TO 10
20  CONTINUE
    J21=1
    NUMBOM(1)=MAX
    AIMLOC(1)=BESTAP(1,1)
    NUMAIM=1
    IF(NGPW.GT.12) GO TO 70
    CALL PROCAL(NUMAIM,AIMLOC,NUMBOM)
    CALL UNION(NUMBOM,NUMAIM,EPSILN,TOTAL)
    GO TO 71
70  CONTINUE
    CALL SIM(NUMAIM,AIMLOC,NUMBOM,TOTAL)
71  CONTINUE
    TMARGIN(1)=1.-TOTAL
100 CONTINUE
    MINNUM=0
    J8=1
    DO 101 I=J8,L
    APHIN(I)=BESTAP(L,I)
    MINBOM(I)=NUMBOM(I)
    MINNUM=MINNUM+MINBOM(I)
101 CONTINUE
    IF(NMAX.EQ.MIN) GO TO 110
    GO TO 10
110 CONTINUE
    NGP=N6
    IF(N5.EQ.0) NGP=1
    RETURN
  END

```

Subroutine PROCAL

SUBROUTINE PROCAL (NUMAIN, AIMLOC, NUMBOM)

```
C*****
C THIS SUBROUTINE (PROCAL) CALCULATES THE PROBABILITIES
C OF A WEAPON LANDING IN A SPECIFIC SECTION GIVEN AN
C AIM POINT STRATEGY (THE NUMBER OF AIM POINTS, THEIR
C LOCATION, AND THE NUMBER OF WEAPONS TARGETED FOR EACH
C SPECIFIC AIM POINT). THIS INFORMATION IS STORED IN
C PROSPG BY GROUP NUMBER, AIM POINT NUMBER, AND SECTION
C NUMBER WITHIN THIS GROUP.
C THE FOLLOWING VARIABLES ARE:
C   1. SECBON=AN ARRAY THAT HAS THE BOUNDARIES (IN TERMS OF
C      STANDARD DEVIATIONS) OF THE SECTIONS
C   2. SECFRA=AN ARRAY THAT HAS THE BOUNDARIES OF THE LAST
C      SECTION IN A GROUPING. THESE BOUNDARIES ARE NOT
C      A MULTIPLE OF STEP AND MUST BE COMPUTED SEPARATELY
C   3. PROSEC=AN ARRAY WHICH HAS THE PROBABILITIES OF
C      A WEAPON LANDING IN THAT SECTION. IT USES
C      AN IMSL SUBROUTINE CALLED MDNOR TO COMPUTE THIS
C      PROBABILITY.
C   4. PROFAC=AN ARRAY OF THE PROBABILITIES A WEAPON LANDS IN THE
C      SECTION OF A SPECIFIC GROUP. IT ALSO USES MDNOR (SEE 3)
C*****
      DIMENSION AIMLOC(20), NUMBOM(20), SECBON(20,20), PROSEC(20,20)
      DIMENSION PROFAC(20,20), DAMLOC(20), SECFRA(20,20)
      COMMON PROSPG(12,20,20)
      COMMON/JOHN/ SECBON, PROSEC, PROFAC, DAMLOC, SECFRA
      COMMON/FIXED/RUNWID, DAMRAD, STADEV, TOWID, PROCUT
      COMMON/GRP/NSPG, NSPW, NGPW, STEP, FRAI, ERUNWI, ERWID
C*****
```

```

DO 1 I=1,NUMAIM
DAMLOC(I)=AIMLOC(I)+DAMRAD
DO 1 J=1,NSPW
X=J-1
SECBON(I,J)=(X*STEP-DAMLOC(I))/STADEV
1 CONTINUE

C DETERMINE THE PROBABILITY OF A WEAPON IMPACTING WITHIN
C EACH SECTION FOR EACH AIM POINT

DO 2 I=1,NUMAIM
NSPWM1=NSPW-1
DO 2 J=1,NSPWM1

C NEED TO COMPUTE PROBABILITY FROM -INF TO LEFT BOND (BOND L)

CALL MDNCR(SECBON(I,J),BOND L)

C NEED TO COMPUTE PROBABILITY FROM -INF TO RIGHT BOND (BOND R)

K=J+1
CALL MDNCR(SECBON(I,K),BOND R)

C PROBABILITY FOR THE SECTION IS THE DIFFERENCE BETWEEN THE
C THE LEFT BOND AND THE RIGHT BOND (PROSEC)

PROSEC(I,J)=BOND R-BOND L
2 CONTINUE

C CALCULATE THE FRACTION OF THE TOWID NOT ACCOUNTED FOR PREVIOUSLY

DO 4 I=1,NUMAIM
DO 4 J=1,NGPW
X=J
SECFRA(I,J)=((X+FRAI)*STEP+TOWID+DAMRAD-DAMLOC(I))/STADEV
IF(((X+FRAI)*STEP+DAMRAD+TOWID).GT.ERWID)
CSECFRA(I,J)=(ERUNWI-DAMLOC(I))/STADEV
4 CONTINUE
DO 6 K=1,NUMAIM
DO 6 I=1,NGPW
J=NSPG+I-1
CALL MDNCR(SECBON(K,J),BOND L)
CALL MDNCR(SECFRA(K,I),BOND R)
PROFAC(K,I)=BOND R-BOND L
6 CONTINUE

C FORM AN 3D ARRAY OF THESE PROBABILITIES WITH 1D=GROUP NUMBER
C 2D=AIM POINT NUMBER, 3D=SECTION NUMBER (PROSPG)

DO 3 J=1,NGPW
II=NSPG+J-1
DO 3 L=1,NUMAIM
DO 3 I=J,II
K=I-J+1
IF(K.LT.NSPG) PROSPG(J,L,K)=PROSEC(L,I)
IF(K.EQ.NSPG) PROSPG(J,L,K)=PROFAC(L,J)
3 CONTINUE
RETURN
END

```

Subroutine UNION

SUBROUTINE UNION(NUMBOM,NUMAIN,EPSILN,TOTAL)

C*****

C THIS SUBROUTINE (UNION) USES THE EVENT COMPOSITION
C METHOD TO CALCULATE THE PROBABILITY OF NOT COMPLETING
C THE CUT. ONE MINUS THIS NUMBER (TOTAL) THEN GIVES
C THE PROBABILITY OF MAKING THE CUT WITH THE GIVEN AIM POINT
C STRATEGY.

C THE FOLLOWING VARIABLES ARE:

- C 1. T=AN ARRAY IN WHICH THE SECTIONS IN A GROUP ARE ADDED
C UP TO DETERMINE THE PROBABILITY FOR THAT GROUP
- C 2. INDX=AN ARRAY WHICH IS USED AS A POINTER TO DETERMINE
C WHICH GROUP IS BEING USED AND ALSO HOW MUCH OF THAT
C GROUP IS TO BE USED IN THE CALCULATION
- C 3. SUM=AN ARRAY WHICH IS COMPRISED OF THE PROBABILITIES
C OF THE COMBINATIONS OF THE GROUPING, I.E. SUM(1) IS
C THE SUM OF THE PROBABILITIES OF ALL GROUPING TAKEN
C ONE AT A TIME. SUM(2) IS THE SUM OF THE INTERSECTIONS
C OF ALL GROUPINGS TAKEN TWO AT A TIME, ETC.
- C 4. EPSILN=A CONVERGENCE CRITERIA ON THE CHANGES IN THE
C PROBABILITY OF MAKING THE CUT.
- C 5. TOTSUM=THE SUMMATION OF ALL SUM(I)

C*****

DIMENSION NUMBOM(20),INDX(20),T(20),SUM(20)

COMMON PROSPG(12,20,20)

COMMON/INDICE/INDX

COMMON/FIXED/RUNWID,DAMRAD,STADEV,TOWID,PROCUT

COMMON/GRP/NSPG,NSPM,NGPW,STEP,FRAI,ERUNWI,ERWID

C*****

```

TOTSUM=0.
DO 101 K=1,NGPW
SUM(K)=0.
K2=K+1
DO 11 I1=1,K2
INDX(I1)=I1
11 CONTINUE
INDX(K)=INDX(K)-1
KK=K
10 INDX(KK)=INDX(KK)+1
J=NGPW-(K-KK)
IF(INDX(KK).GT.J) GO TO 100
IF(KK.EQ.K) GO TO 20
M=KK+1
DO 15 KKK=M,K
INDX(KKK)=INDX(KK)+(KKK-KK)
15 CONTINUE
KK=K
20 PROD=1.
DO 30 L=1,NUMAIM
T(K)=0.
DO 25 I=1,NSPG
T(K)=T(K)+PROSPG(INDX(K),L,I)
25 CONTINUE
S=1-T(K)
MM=K-1
IF(MM.EQ.0) GO TO 29
DO 27 IK=1,MM
M1=IK+1
IN=INDX(M1)-INDX(IK)
LIM=MINO(NSPG,IN)
T(IK)=0.
DO 28 I=1,LIM
T(IK)=T(IK)+PROSPG(INDX(IK),L,I)
28 CONTINUE
S=S-T(IK)
27 CONTINUE
29 CONTINUE
PROD=PROD*(S*NUMBOM(L))
30 CONTINUE
SUM(K)=SUM(K)+PROD
GO TO 10

```

```

100  KK=KK-1
      IF (KK.GT.0) GO TO 10
      B=K+1
      Z=B/2
      J5=(K+1)/2
      A=Z-J5
      J6=0
      J7=J6-1
      IF (A.GT.0) SUM(K)=J7*SUM(K)
      TOTSUM=TOTSUM+SUM(K)
      IF (K.EQ.1) GO TO 200
      U=ABS(SUM(K-1))
      V=ABS(SUM(K))
      W=U-V
      TOLAN=ABS(W)
      IF (TOLAN.LT.EPSILN) GO TO 300
200  CONTINUE
101  CONTINUE
300  TOTAL=TOTSUM
      RETURN
      END

```

Subroutine SIM

SUBROUTINE SIM(NUMAIM,AIMLOC,NUMBOM,TOTAL)

C*****

C THIS SUBROUTINE (SIM) USES A MONTE CARLO SIMULATION
C TO DETERMINE THE PROBABILITY OF NOT MAKING THE CUT.
C THIS SUBROUTINE IS USED WHENEVER THE NGPW EXCEEDS 12.
C IF NGPW EXCEEDS 12 THIS SUBROUTINE IS "QUICKER" THAN
C UNION.

C THE FOLLOWING VARIABLES ARE:

- C 1. RANF(DUM)=A RANDOM NUMBER GENERATOR
- C 2. GSNHL=IMSL SUBROUTINE WHICH RETURNS A VECTOR OF
C NORMAL (0,1) STANDARD DEVIATES.
- C 3. VSTRA=IMSL SUBROUTINE WHICH SORTS A VECTOR AND
C PLACES THE VALUES IN ASCENDING ORDER. THE FIRST
C NUMBER IN THE RETURNED VECTOR IS THE SMALLEST.
- C 4. NITERA=THE NUMBER OF ITERATIONS FOR THIS SIMULATION

C*****

DIMENSION NUMBOM(20),AIMLOC(20)
DIMENSION HIT(50),R1(1)
COMMON/SIMS/HIT,R1
COMMON/FIXED/RUNWID,DAMRAD,STADEV,TOWID,PROCUT
COMMON/ITERAT/NITERA
DOUBLE PRECISION DSEED

C*****

```

      DO 100 I=1,50
      HIT(I)=0.0
100   CONTINUE
      E=100000000.
      M=0
      C=0.
      C=C-DAMRAD
      D=DAMRAD+RUNWID
      CLEAR=2.*DAMRAD+TOWID
      ECLEAR=DAMRAD+TOWID
      NUMB=0
      DO 1 I=1,NUMAIM
      NUMB=NUMB+NUMBOM(I)
1     CONTINUE

```

```

      I=0
      N=NITERA
2     CONTINUE
      I=I+1
      IF(I.EQ.N+1) GO TO 30
      X=RANF(DUM)
      IX=X*E
      IF(IX.EQ.0) IX=585285124
      DSEED=IX
      N2=0
      DO 3 J=1,NUMAIM
      N1=NUMBOM(J)
      DO 3 J1=1,N1
      N2=N2+1
      I2=1
      CALL GGNML(DSEED,I2,R1)
      HIT(N2)=AIMLOC(J)+STADEV*R1(I2)
3     CONTINUE
      CALL VSRTA(HIT,NUMB)

```

C
C DOES A WEAPON LAND ON THE EFFECTIVE RUNWAY

```

      N4=0
      N5=0
      N6=0
      DO 8 L=1,NUMB
      IF(HIT(L).LT.C) N4=N4+1
      IF(HIT(L).GT.D) N5=N5+1
8     CONTINUE
      IF(N4.EQ.NUMB) GO TO 20
      IF(N5.EQ.NUMB) GO TO 20
      N6=N4+N5
      IF(N6.EQ.NUMB) GO TO 20
      N4=N4+1

```

C IS THE FIRST WEAPON CLOSE TO THE RUNWAY EDGE

```

      DIS=0.
      DIS=0.-HIT(N4)
      IF(ABS(DIS).GT.ECLEAR) GO TO 20

```

C IS THE "LAST" WEAPON CLOSE TO THE RUNWAY EDGE

```

      DIS=0.
      N9=NUMB-N5
      DIS=RUNWID-HIT(N9)
      IF(ABS(DIS).GT.ECLEAR) GO TO 20

```

C CHECK ON THE DISTANCE BETWEEN ADJACENT IMPACT POINTS ON THE EFFECTIVE
C RUNWAY

```

DIS=0.
IF((N4+N5).EQ.(NUMB-1)) GO TO 4
N8=NUMB-N5-1
IF(N4.EQ.N8) GO TO 4
IF(N8.EQ.0) DIS=HIT(N4)-HIT(N9)
IF(N8.EQ.0) GO TO 4
DO 7 J=N4,N8
DIS=HIT(J+1)-HIT(J)
IF(ABS(DIS).GT.CLEAR) GO TO 20
7 CONTINUE
4 CONTINUE
IF(ABS(DIS).GT.CLEAR) GO TO 20
GO TO 2
20 CONTINUE
Z=0.
M=M+1
GO TO 2
30 CONTINUE
A=M
B=I-1
TOTAL=A/B
RETURN
END

```

Subroutine RUNPRO

SUBROUTINE RUNPRO(Y,NCUT,BNDLOW,BNDUP)

C*****

C THIS SUBROUTINE (RUNPRO) CALCULATES THE INTERVAL FOR THE PROBABILITY
C OF DESTROYING THE RUNWAY AT A CONFIDENCE LEVEL OF .99

C THE FOLLOWING VARIABLES ARE:

- C 1. ESTX=THE ESTIMATE OF THE PROBABILITY OF DESTROYING THE RUNWAY
- C 2. EVARY=THE ESTIMATE OF THE VARIANCE IN MEAN FOR ONE CUT
- C 3. ESTYSQ=THE MEAN SQUARED OF A CUT
- C 4. EVARX=ESTIMATE OF THE VARIANCE FOR THE RUNWAY DESTRUCTION
- C 5. STANDX=THE STANDARD DEVIATION OF THE RUNWAY DESTRUCTION
- C 6. BNDLOW=THE LOWER BOUND ON THE INTERVAL
- C 7. BNDUP=THE UPPER BOUND ON THE INTERVAL

C*****

COMMON/ITERAT/NITERA

C*****

X1=NITERA
X2=NCUT
ESTX=Y**X2
EVARY=Y*(1.-Y)/X1
ESTYSQ=Y**2.
EVARX=((EVARY+ESTYSQ)**X2)-(ESTX**2.)
STANDX=EVARX**.5
BNDLOW=ESTX-3.*STANDX
BNDUP=ESTX+3.*STANDX
RETURN
END

APPENDIX D

Derivation of the Mean and Variance of \hat{P} , the Estimated Probability of Destroying the Runway With the "Best" Aim Point Strategy

Derivation of the Mean and Variance
of \hat{P} , the Estimated Probability of
Destroying the Runway With the
"Best" Aim Point Strategy

One of the basic assumptions used in this model is that the overall problem of destroying the runway can be expressed as N identical and independent subproblems (see Note on page 8). One also knows the distribution of the estimates for the actual value of p , the probability of making a cut with a given aim point strategy. If one lets: 1) Y be the random variable which denotes the estimates of p ; 2) $F_Y(y)$ is the distribution of Y ; and 3) Y_1, Y_2, \dots, Y_N are i.i.d. from $F_Y(y)$. Thus,

$$E[Y_1 Y_2 \dots Y_N] = E[Y_1] E[Y_2] \dots E[Y_N] \quad (D-1)$$

where $E[Y]$ is the expectation of the random variable Y (Ref 11: 179).

Since the subproblems are identical, i.e. $E[Y_1] = E[Y_2] = \dots = E[Y_N]$, Eq (D-1) becomes

$$E[Y_1 Y_2 \dots Y_N] = (E[Y_1])^N \quad (D-2)$$

Also, according to Theorem 3.4 (Ref 11: 90)

$$V[A] = E[A^2] - \mu^2 \quad (D-3)$$

where $V[A]$ is the variance of the random variable A and μ is the $E[A]$. Thus,

$$V[Y_1 Y_2 \dots Y_N] = E[(Y_1 Y_2 \dots Y_N)^2] - (E(Y_1 Y_2 \dots Y_N))^2 \quad (D-4)$$

Coupling the assumption that the Y s are independent, identically distributed random variables with Eq (D-3)

$$V[Y_1 Y_2 \dots Y_N] = (E[(Y_1)^2])^N - (E[Y_1])^{2N} \quad (D-5)$$

According to Mendenhall and Scheaffer, Y_1 can be viewed as asymptotically normal with mean p and variance $p(1-p)/M$, where M is the number of simulated attacks (Ref 11: 256).

Thus,

$$E[Y_1] = \hat{p} \quad (D-6)$$

where \hat{p} is the point estimate of PC for the "best" aim point strategy, and

$$V[Y_1] = \frac{\hat{p}(1-\hat{p})}{M} \quad (D-7)$$

By using Eq (D-4)

$$E[(Y_1)^2] = \frac{\hat{p}(1-\hat{p})}{M} + (\hat{p})^2 \quad (D-8)$$

Thus

$$V[Y_1 Y_2 \dots Y_N] = \left(\frac{\hat{p}(1-\hat{p})}{M} + (\hat{p})^2 \right)^N - (\hat{p})^{2N} \quad (D-9)$$

To construct a 99 percent confidence level on an interval estimate of \hat{p} , one can use $\pm 3\sigma$, the standard deviation, about the point estimate \hat{p} . To accomplish this one needs,

$$\hat{P} = (\hat{p})^N \quad (D-10)$$

and

$$\hat{\sigma} = ((\frac{\hat{p}(1-\hat{p})}{M} + (\hat{p})^2)^N - (\hat{p})^{2N})^{1/2} \quad (D-11)$$

(See Appendix C, Subroutine RUNPRO for the coding of this derivation.)

Vita

John Calvin Pemberton was born on 18 February 1951 in Fukuoka, Japan. He graduated from Artesia High School in Artesia, N.M. in 1969. Immediately following his high school graduation, John entered the United States Air Force Academy. On 6 June 1973, he graduated from USAFA with a Bachelor of Science degree in Basic Sciences and received a regular commission in the United States Air Force. He completed navigator training and received his wings in April 1974. After completing C-130 upgrade training at Little Rock AFB, Ar., he was assigned to the 37th TAS, 316th TAW, Langley AFB, Va. During the two and a half years at Langley, he served as a C-130 navigator and as an instructor navigator. John was then reassigned to the 345th TAS, 374th TAW, Yokota AB, Japan where he advanced to the level of squadron flight examiner. While assigned to the 345th TAS, he attended Squadron Officer School at Maxwell AFB, Al., where he received the Commandant's Trophy for being the top student in his class, 1978D. In June 1979, he entered the School of Engineering, Air Force Institute of Technology.

Permanent Address: 1801 Briscoe Ave
Artesia, NM 88210

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFIT/GOR/OS/80D-6	2. GOVT ACCESSION NO. AD-A094 807	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A GENERALIZED COMPUTER MODEL FOR THE TARGETING OF CONVENTIONAL WEAPONS TO DESTROY A RUNWAY		5. TYPE OF REPORT & PERIOD COVERED MS thesis
7. AUTHOR(s) John C. Pemberton Capt USAF		6. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS Air Force Institute of Technology (AFIT/EN) Wright-Patterson AFB, Ohio 45433		8. CONTRACT OR GRANT NUMBER(s)
11. CONTROLLING OFFICE NAME AND ADDRESS		10. PROGRAM ELEMENT PROJECT, TASK AREA & WORK UNIT NUMBERS
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE December 1980
		13. NUMBER OF PAGES 114
		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report) <i>Laurel A. Lampela</i> LAUREL A. LAMPELA, 2dLt, USAF, Air Force Institute of Technology (ATCI) Senior Director, Public Affairs, Wright Patterson AFB, OH 45433		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Probability of Kill Targeting Schemes "MASSIVE" Event-Composition Probability Circular Error Probable Runway Destruction Effective Explosive Weight Conventional Weapons Airfield Assessment Program		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) A generalized computer model for determining where to target conventional weapons against a non-reinforced concrete runway is developed. This development consists of taking the overall problem, runway destruction, and dividing it into a number of identical, stochastically independent subproblems. Then one of these subproblems is used to determine an aim		

DD FORM 1 JAN 73 1473

EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

Block 20:

point strategy which reduces the number of weapons required to achieve the level of runway destruction desired.

In order to solve the subproblem, a search algorithm is developed. Along with this search algorithm, two methods for approximating the level of destruction, given an aim point strategy, are investigated. Finally, the entire model is verified and "validated" by a number of statistical tests. Also a comparison of this model to the Airfield Assessment Program, an Air Force Armament Laboratory model, is provided. Using the model's results for the data investigated, an equation relating circular error probable and minimum launch window's width to the number of weapons employed is derived.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

ND
ATE